

**SOLUCIÓN AL PROBLEMA DE RUTEO DE VEHICULOS CON VENTANAS DE  
TIEMPO, FLOTA HETEROGÉNEA Y MULTIPLES PRODUCTOS MEDIANTE  
UNA METAHEURÍSTICA BASADA EN EL FOTOTAXISMO DE LAS  
BACTERIAS.**

**JAIME EDUARDO AMADOR FONTALVO**

**BARRANQUILLA  
UNIVERSIDAD DEL NORTE  
DIVISIÓN DE POSTGRADOS E INVESTIGACIONES EN INGENIERÍA  
MAESTRÍA EN INGENIERÍA INDUSTRIAL  
2012**

**SOLUCIÓN AL PROBLEMA DE RUTEO DE VEHICULOS CON VENTANAS DE  
TIEMPO, FLOTA HETEROGÉNEA Y MULTIPLES PRODUCTOS MEDIANTE  
UNA METAHEURÍSTICA BASADA EN EL FOTOTAXISMO DE LAS  
BACTERIAS.**

**JAIME EDUARDO AMADOR FONTALVO**

**Proyecto de grado presentado como requisito parcial  
para optar al título de Magíster en Ingeniería Industrial  
Área de énfasis: Métodos Cuantitativos**

**Director: Carlos Daniel Paternina Arboleda.  
Ingeniero Industrial, Ph. D.**

**BARRANQUILLA  
UNIVERSIDAD DEL NORTE  
DIVISIÓN DE POSTGRADOS E INVESTIGACIONES EN INGENIERÍA  
MAESTRÍA EN INGENIERÍA INDUSTRIAL  
2012**

Aprobado por la División de Postgrados e Investigaciones en Ingeniería en cumplimiento de los requisitos exigidos para otorgar al título de Magíster en Ingeniería Industrial, área de énfasis en Métodos Cuantitativos.

---

Ing. Carlos D. Paternina A., Ph. D.  
Director del Proyecto

---

Víctor Cantillo Maza, Ph. D.  
Jurado

---

Daladier Jabba Molinares Ph. D.  
Jurado

Barranquilla, Noviembre de 2011

A Yamile y Jaime, mis padres,  
por haberme dado el precioso  
regalo de traerme a la vida y ser  
los principales gestores de mi  
formación tanto personal como  
profesional. A mi novia, a mi  
familia, mis amistades más  
cercanas y mis compañeros de  
clase por brindarme siempre su  
apoyo incondicional.

Jaime Eduardo Amador  
Fontalvo



## **AGRADECIMIENTOS**

El autor desea expresar sus agradecimientos a:

CARLOS DANIEL PATERNINA ARBOLEDA, Ph. D., Profesor de la escuela de la Escuela de Negocios, Universidad del Norte, y director del proyecto.

ANGEL LEON GONZALES ARIZA, Ph. D., Profesor tiempo completo del departamento de Ingeniería Industrial, Universidad del Norte.

Mi padre JAIME EDUARDO AMADOR GOMEZ, por su apoyo, sugerencias y consejos referentes a programación, que me permitieron desarrollar dicho trabajo.

Todas las personas y organizaciones que de una u otra manera cooperaron y colaboraron con el desarrollo del presente proyecto de grado.

## TABLA DE CONTENIDO

INTRODUCCIÓN .....	1
1. PLANTEAMIENTO DEL PROBLEMA .....	4
1.1. ANTECEDENTES .....	4
1.1.1. Investigaciones Anteriores Relacionadas con la Temática a Desarrollar. ....	5
1.2. FORMULACIÓN DEL PROBLEMA. ....	8
1.3. JUSTIFICACIÓN.....	12
2. OBJETIVOS DE LA INVESTIGACIÓN. ....	14
2.1. OBJETIVO GENERAL.....	14
2.2. OBJETIVOS ESPECÍFICOS. ....	14
3. ALCANCES Y LIMITACIONES DE LA INVESTIGACIÓN. ....	16
4. MARCO TEÓRICO. ....	18
4.1. PROBLEMA DEL AGENTE VIAJERO.....	18
4.2. PROBLEMA DE ENRUTAMIENTO DE VEHICULOS.....	19
4.2.1. Variantes del VRP. ....	20
4.3. OPTIMIZACIÓN MULTIOBJETIVO. ....	22
4.4. OPTIMIZACIÓN COMBINATORIA. ....	23
4.4.1. Heurísticas de Construcción. ....	23
4.4.2. Métodos de Búsqueda Local. ....	24
4.4.3. Metaheurísticas.....	25
4.5. APRENDIZAJE DE MAQUINA Y APRENDIZAJE REFORZADO .....	26
4.6. HEURISTICA DE OPTIMIZACIÓN MULTIOBJETIVO BASADA EN EL FOTOTAXISMO DE LAS BACTERIAS.....	27
5. PROPUESTA METAHEURISTICA PARA EL MPHVRPTW .....	31
5.1. DESCRIPCIÓN DE LA METAHEURISTICA.....	32
5.1.1. Colonia de bacterias iniciales. ....	32
5.1.2. Reducción del número de rutas empleadas.....	34
5.1.3. Ciclos Bacterianos. ....	37
5.1.4. Giro Bacterias.....	37
5.1.5. Carrera hacia la luz. ....	38
5.1.6. Fisión Binaria. ....	39

5.1.7.	Mutación Espontánea.....	40
5.1.8.	Mutación reversa. ....	41
5.1.9.	Selección Muerte de Bacterias.....	42
5.1.10.	Fotosíntesis. ....	42
5.1.11.	Selección de la mejor solución. ....	43
6.	EXPERIMENTACION Y RESULTADOS.....	44
6.1.	DEFINICIÓN DE PARAMETROS.....	46
6.2.	ANÁLISIS DE RESULTADOS.....	47
6.2.1.	Tiempo computacional.....	53
7.	CONCLUSIONES Y FUTURAS INVESTIGACIONES.....	55
	REFERENCIAS BIBLIOGRAFICAS.....	58
	ANEXOS.....	64



## **LISTA DE TABLAS.**

Tabla 1. Resultados de las corridas para las instancias a evaluar.....	51
Tabla 2. Resumen de resultados obtenidos para las instancias R201 y R209 .....	52
Tabla 3. Comparación de la metodología propuesta con otras heurísticas. ....	53
Tabla 4. Tiempos de ejecución del algoritmo en las instancias seleccionadas.....	53

## **LISTA DE FIGURAS.**

Figura 1. Inserción de un cliente. ....	37
---	----

## **LISTA DE ANEXOS.**

Anexo 1. Formula Giro Bacteria. ....	64
Anexo 2. Análisis de varianza. ....	67
Anexo 3. Instancia R201 para HFMPVRPTW .....	69
Anexo 4. Instancia R209 para HFMPVRPTW .....	71
Anexo 5. Mejor solución encontrada en la instancia R201 .....	73
Anexo 6. Mejor solución encontrada en la instancia R209 .....	78

## INTRODUCCIÓN

Entre las diferentes actividades llevadas a cabo dentro de la una cadena de abastecimiento SC, el transporte juega un rol vital en los costos asociados a la logística. De hecho, el proceso de transporte está implicado en todas las etapas de los sistemas de producción y distribución, y representa un componente relevante (generalmente entre el 10% y 20%) del costo final de los productos. (Toth y Vigo 2002). Esto refleja el porqué los tomadores de decisiones en las compañías buscan las mejores estrategias para la planificación de las operaciones relacionadas con el transporte de sus productos.

Ahora, en la gestión del transporte, se plantean diferentes objetivos para realizar reducciones significativas en cuanto a costos, donde los más comunes son los siguientes: balanceo de rutas con relación a los tiempos de viajes y carga del vehículo, cobertura de todos los clientes con el número menor de vehículos, reducción del costo total de transporte relacionados con la distancia total recorrida (o el tiempo total de viaje) y los costos fijos de cada vehículo utilizado, penalizaciones adquiridas por el no servicio o servicio tardío a los clientes que se deben visitar, y cualquier combinación de estos. Sin embargo los encargados de la toma de decisión del transporte se enfrentan a diferentes restricciones operativas, a las características que poseen los clientes y a las características de los vehículos, las cuales son elementos determinantes para el buen diseño de la estrategia de transporte que se necesite implementar. Con relación a las características de los clientes se tiene: la localización, la cantidad de productos a entregar o a recoger, intervalos de tiempo dentro de los cuales puede ser atendido, tipo de carga y de vehículos que se pueden utilizar para esta, y tiempos requeridos para recoger o entregar los productos a dicho cliente (tiempo de servicio). Para las características de los vehículos se pueden destacar: diferentes tipos de vehículo con relación a características técnicas tales como capacidad de carga, velocidad, etc.; depósito

inicial del vehículo y la posibilidad de que al finalizar la ruta llegue al depósito inicial o a otro distinto; costos asociados a la utilización de un vehículo en particular y posibles subdivisiones del vehículo en compartimientos, donde cada uno de estos puede llevar una cantidad y un tipo de carga en especial.

Para las restricciones operativas caben mencionar: cantidad de vehículos a utilizar, límite de carga de cada uno de los vehículos; los clientes en la ruta solamente requieren ya sea la entrega o recolección, o quizás ambas posibilidades; restricciones de precedencia, en donde la secuencia en que se deban visitar a los clientes es fija o una ruta en la cual atiende a dos tipos de clientes debe atender a un tipo antes que a el otro y restricciones de distancia donde un vehículo no puede superar cierta cantidad de distancia recorrida (o tiempo empleado).

En los problemas de ruteo de vehículos mencionados anteriormente, las personas encargadas de tomar la decisión del enrutamiento, algunas veces se basan en lo que han experimentado durante el día a día de su jornada laboral o simplemente en lo que ellos creen que sería la mejor alternativa para dar solución a dicha situación. Sin embargo algunas de soluciones se encuentran muy lejos de la óptima, generando un impacto negativo en la efectividad operacional y económica de toda la SC. Dichos problemas de ruteo que poseen clientes y vehículos con diferentes características, además de las restricciones operativas, resultan ser complejos y la solución que se busca puede ser exactamente la ideal o una aproximación a esta. En dicha búsqueda es cuando se habla de optimización.

Para estos problemas, se pueden aplicar métodos de optimización matemática que dan una solución puntual, sin embargo esta alternativa no resulta ser buena en problemas que implican múltiples objetivos, en lo referente a tiempo computacional. Con lo anterior se puede pensar en resolver dichos problemas mediante heurísticas, las cuales presentan un esquema computacional que pueden alcanzar el óptimo en un tiempo computacional adecuado. Pero la búsqueda de soluciones con las heurísticas en algunos momentos puede presentar ciertos inconvenientes, por lo tanto se han creado las llamadas metaheurísticas

que no presentan los inconvenientes de las heurísticas y que pueden obtener una solución aproximativa prácticamente a cualquier tipo de problema combinatorio.

Las metaheurísticas, que básicamente están basadas en diferentes fenómenos de la naturaleza, han sido utilizadas ampliamente y entre las más divulgadas se encuentran el Recocido Simulado (SA), Colonia de Hormigas (ACO), Búsqueda Tabú (TS), Algoritmos Genéticos (GA), Algoritmos SWARM (PSO) y los híbridos realizados entre estas. Aunque en los últimos años se han desarrollado nuevas metaheurísticas, existe una basada en el fototaxis de las bacterias (GLSBA) (Gómez Vizcaíno 2009) y que hasta el momento no se ha aplicado en la solución de muchos problemas multiobjetivo.

La siguiente investigación está encaminada a dar solución a un problema de enrutamiento de vehículos mediante la metaheurística GLSBA. Dicho problema ha sido estudiado con anterioridad mediante una heurística de dos fases que emplea TS y ACO (De La Cruz, y otros 2011) (De La Cruz Hernández 2003) y la metodología se probará con instancias que ya fueron modificadas por el mismo autor, debido a que para dicho problema no se han desarrollado suficientes investigaciones y no existen instancias definidas por otros autores.

## **1. PLANTEAMIENTO DEL PROBLEMA**

### **1.1. ANTECEDENTES.**

La logística implica brindarle al cliente el producto que necesita de acuerdo a los diferentes requerimientos que este tenga, los cuales abarcan cantidad, calidad, lugar y tiempo, y con el fin de que lo anterior suceda, debe implementarse una estrategia de transporte adecuada. Esta situación no es ajena a nuestro entorno y podemos ver en nuestro país los esfuerzos de diferentes compañías por cumplir y superar las expectativas de sus clientes al aplicar una buena gestión.

Cada una de estas compañías debe enfrentar los diferentes escenarios que se le pueden presentar de acuerdo a las características de los clientes a visitar, las características de la flota de vehículos que posee y las restricciones operativas, temática que ya fue discutida anteriormente. En nuestro país es frecuente que muchos clientes posean características de ventanas de tiempo, y un ejemplo claro de esto pueden ser los supermercados, puesto que al tener una gran cantidad de productos (carnes, lácteos, frutas, verduras, ropa, etc.) de los que se abastecen de diferentes proveedores, no pueden (o les resultaría muy costoso) realizar la recepción de todos los productos al mismo tiempo. Por lo tanto, a cada proveedor se le impone un intervalo de tiempo el cual debe cumplir. Estas situaciones son los llamados problemas de ruteo de vehículos con ventanas de tiempo (VRPTW por sus siglas en inglés).

Siguiendo con el ejemplo del supermercado, cada proveedor no solo se enfrenta con el intervalo de tiempo impuesto, sino que además debe lidiar con la cantidad y tipos de productos que debe hacer entrega (o recoger según sea el caso). Para lo anterior, los proveedores cuentan con diferentes tipos de vehículos cada uno con características técnicas propias, siendo la más común la capacidad de carga y esto conlleva a los problemas llamados problemas de ruteo de vehículos con flota heterogénea (HFVRP por sus siglas en inglés). El vehículo que visitará el supermercado no solo lo hará con este sino que debe cumplir con otros pedidos por lo que se debe garantizar que la capacidad sea la adecuada

para cumplir con los diferentes tipos de productos (generalmente cada uno con un volumen y peso determinado) solicitados en las cantidades requeridas. Estos problemas con conocidos en la literatura como problemas de ruteo de vehículos con múltiples productos (MPVRP).

En la vida real, quienes están encargados de realizar el ruteo de vehículos se encuentran con los problemas mencionados anteriormente y la mayor parte del tiempo resultan ser una mezcla de los tres tipos mencionados. Al revisar la literatura, se encuentran algunos trabajos que dan diferentes soluciones a los VRPTW, HFVRP y MPVRP por separado, pero han sido muy pocos los que tratan la mezcla de esta terna.

#### **1.1.1. Investigaciones Anteriores Relacionadas con la Temática a Desarrollar.**

Estudios previos del problema de ruteo de vehículos con ventanas de tiempo (VRPTW), incluyen tanto algoritmos de optimización como aproximaciones heurísticas, donde estas últimas pueden definirse en dos áreas: construcción de rutas y mejoramiento de rutas.

Con relación a la construcción de rutas para el VRPTW, existe el conocido método de ahorros (Clarke y Wright 1964) el cual ha presentado modificaciones (Baker y Schaffer 1989), (Solomon 1987) (Van Landeghem 1988), donde cada una de estas modificaciones poseen características particulares para la construcción de la ruta. Por otro lado, existen técnicas basadas en la formación de clústeres, (Guillet y Miller 1974) y técnicas de inserción (Potvin y Rousseau 1993) (Russell 1995).

Las técnicas de mejoramiento de rutas se basan en el concepto de vecindades, donde estas son generadas mediante diferentes estructuras como son 2-opt y Or-opt, con ciertas modificaciones (Potvin y Rousseau 1995) a fin de que cumplan con las características del VRPTW. Otros operadores de vecindades tales como  $\lambda$ -interchange y exchange también se han aplicado para dar solución a los VRPTW. En resumen, una revisión de las diferentes heurísticas para resolver el VRPTW está consignada en el trabajo realizado por Tan y otros (2001).



Las metaheurísticas son herramientas sumamente útiles a un alto número de problemas de optimización combinatoria, y estas se han usado en los últimos años con el fin de obtener mejores soluciones para el VRPTW, donde las más conocidas son: Recocido Simulado (SA), Colonia de Hormigas (ACO), Búsqueda Tabú (TS) y Algoritmos Genéticos (GA). Para el SA, se han desarrollado tres métodos basados en SA para la solución del VRPTW (Russell y Chiang 1996), además de otra solución alternativa basada en esta metaheurística (Thangiah, Osman y Sun 1995). Con respecto al ACO se desarrolló una solución para el VRPTW basado en un sistema de múltiple ACO (Gambardella, Taillard y Agazzi 1999), y posteriormente otras propuestas para el VRPTW basadas en el comportamiento de las hormigas (Tan, y otros 2005).

En cuanto a TS, se destacan las heurísticas de búsqueda tabú paralela para la solución de los VRPTW (Badeau, y otros 1997) (Garcia, Potvin y Rousseau 1994) (Schulze y Fahle 1999). Entre otros trabajos basados en TS se pueden encontrar: heurísticas TS unificadas para el VRPTW al igual que para dos generalizaciones de esta (con diferentes depósitos o bodegas MDVRPTW y por periodos de tiempo PVRPTW) (Cordeau, Laporte y Mercier 2001); metaheurística TS para el resolver el VRPTW con ventanas de tiempo “suaves”, es decir donde se aplica una penalización si no se cumple con la restricción de la ventana de tiempo (Russell y Chiang 2004); solución al problema de VRPTW con entregas separadas VRPTWSD (Ho y Haugland 2004).

Para los algoritmos genéticos GA, existe uno donde se utiliza la técnica de Agrupar Primero – Enrutar Después (Thangiah 1995) al igual que un GA llamado Genetic Routing System (GENEROUS) donde los operadores evolutivos solo se aplican a las soluciones factibles y no sobre una codificación de estas (Potvin y Bengio 1996). Otros autores examinan los diferentes tipos de enfoque desarrollados para la implementación de los GA en la solución de los VRPTW (Bräysy y Gendreau 2001), han propuesto algoritmos genéticos que no parten de un conjunto de soluciones sino de una única solución a la cual se le aplican los diferentes operadores (Bräysy y Dullaert 2003) y otros desarrollaron una propuesta de un GA híbrido para solucionar el problema en cuestión, el cual se basa en la evolución de dos poblaciones de soluciones (Berger y Barkaoui 2004). Existen artículos

donde se realiza una revisión más detallada de GA para el VRPTW. (Bräysy, Dullaert y Gendreau 2004a) (Bräysy y Gendreau 2005b).

En los últimos años, las propuestas híbridas se han vuelto muy populares puesto que la combinación de estas han dado mejores resultados que una sola meta heurística. En uno de estos híbridos se empleó una combinación entre TS y GA para solucionar el VRPTW encontrándose un desempeño superior al de una sola meta-heurística (Chin, Kit y Lim 1999). Otros híbridos combinan hasta tres metaheurísticas tales como GA, TS Y SA (Thangiah, Osman y Sun 1994) y también existen trabajos donde se combinan estrategias evolutivas y TS para minimizar el número de vehículos y la distancia total para el VRPTW (Homburger y Gehring 2005). Por otro lado, se ha creado una heurística de dos fases para resolver los VRPTW, usando un GA y formulación de partición de conjuntos.(Alvarenga, Mateus y De Tomi 2007), al igual que un algoritmo híbrido de dos etapas, el cual combina el SA y búsqueda amplia de vecindad para el Pickup and Delivery VRPTW con múltiples vehículos (PDVRPTW) (Bent y Van Hentenryck 2006). Para la solución del VRP con restricción de capacidad (CVRP) se desarrolló un híbrido SA basado en el *vecino más cercano* para minimizar el costo de la flota heterogénea y maximizar la utilización de capacidad (Tavakkoli-Moghaddam, Safaei y Gholipour 2006).

El VRPTW ha presentado diferentes variantes, que algunas ya han sido mencionadas como son el PDVRPTW y el CVRP. Sin embargo para esta investigación cabe mencionar algunas con las que se guarda una estrecha relación. La primera de estas es el VRPTW con flota heterogénea de vehículos (HFVRPTW), donde la diferencia radica en que la flota de los vehículos utilizados posee características diferentes, tales como capacidad de carga, velocidad, costo de adquisición, etc. Entre los trabajos relacionados con esta variante se destaca un método de construcción de rutas para esta variante donde los vehículos difieren por capacidad y costo de adquisición (Liu y Shen 1998), mientras que existe una alternativa basada en el método de generación de columnas para resolver el VRP con flota heterogénea (HFVRP) (Choia y Tchab 2007). Por otro lado, se ha desarrollado una propuesta de solución al VRPTW con flota heterogénea de vehículos y varios depósitos (MDHFVRPTW) mediante una heurística de tres fases a fin de minimizar costos de

utilización de los vehículos, costos de tiempo y distancia recorrida, y costos por tiempo de servicio y espera.(Dondo y Cerdá 2007).

La otra variante relacionada con esta investigación es el VRPTW pero con diferentes tipos de producto (MPVRPTW), de la cual hasta el momento no existen muchos trabajos relacionados con esta variante. Solo Dantzing y Ramser en el año de 1959, discuten brevemente esta variante pero sin mencionar técnicas para su solución. (Aronson 1995)

Hasta el momento, solo se encuentra una metodología para solucionar el problema de ruteo de vehículos con ventanas de tiempo, flota heterogénea y múltiples productos (HFVRPTWMP), a través de una metodología de dos fases en las que emplea TS, ACO y heurísticas de búsqueda local (De La Cruz Hernández 2003) (De La Cruz, y otros 2011). Puesto que no existen instancias en las cuales se consideren mezclas de productos con pesos y volúmenes conocidos, el autor realizó las modificaciones pertinentes al conjunto de problemas generados por Marius Solomon.<sup>1</sup>

En cuanto a metaheurísticas, recientemente se desarrolló un algoritmo de optimización multiobjetivo basado en el fototaxismo de las bacterias (Gómez Vizcaíno 2009), el cual será utilizado para dar solución al VRPTW que se ha planteado en esta investigación.

La siguiente investigación está encaminada en el desarrollo de una alternativa de solución al HFVRPTWMP mediante un algoritmo, basado en heurísticas de búsqueda local y la metaheurística GLSBA, donde se busca reducir la cantidad de vehículos utilizados y la distancia total recorrida por estos.

## **1.2. FORMULACIÓN DEL PROBLEMA.**

El problema a resolver consiste en realizar el enrutamiento de una flota de vehículos que tienen una capacidad restringida heterogénea (medida en peso y volumen de carga) hacia un número definido de puntos de demanda (clientes). Estos clientes se caracterizan por

---

<sup>1</sup> Estos problemas se pueden obtener de la página personal de Marius Solomon. VRPTW Benchmark Problems, URL: <http://web.cba.neu.edu/~msolomon/problems.htm>

tener un intervalo de tiempo o *ventana de tiempo*, la cual no puede ser violada, es decir el vehículo no puede llegar después de la cota superior de tiempo, aunque la llegada por debajo de la cota inferior es permitida e implica realizar una espera. Por otro lado, cada cliente tiene cierta cantidad de productos, medidos en peso y volumen, que deben ser recogidos por uno y solo un vehículo, es decir, cada vehículo visita un cliente una sola vez y cada cliente no puede ser visitado por dos o más vehículos. En cuanto a los vehículos, todos parten desde una bodega o depósito inicial y deben regresar a esta en un intervalo de tiempo dado, que para este caso es el mismo intervalo para todos los vehículos, además de que no se puede exceder la capacidad de carga que cada uno posee. La idea es dar total cobertura a los clientes, cumpliéndose todo lo mencionado anteriormente, utilizando el menor número de vehículos y empleando la menor distancia recorrida posible. Matemáticamente, lo anterior se puede expresar de la siguiente manera:

Sea  $N = \{1, 2, \dots, n\}$  el conjunto de clientes y  $K$  el conjunto de vehículos disponibles para ser programados. Considere el grafo  $G^k = (V^k, A^k)$ ,  $\forall k \in K$ , cada uno de ellos consiste en un conjunto  $V^k$  de nodos y un conjunto  $A^k$  de arcos. El conjunto  $V^k$  de nodos consiste en  $N \cup \{0\}$  donde 0 representa la bodega o depósito origen y destino del vehículo  $k$ ,  $k \in K, 1 \leq k \leq K$ . El conjunto  $A^k$  contiene todos los arcos factibles, el cual es subconjunto de  $V^k \times V^k$ . Sea  $p_i$  y  $v_i$  el peso y volumen de la demanda del cliente  $i$ ,  $\forall i \in N \cup \{0\}$  donde  $p_0 = v_0 = 0$ , y existe una capacidad límite en peso  $P_k$  y en volumen  $V_k$ ,  $\forall k \in K$ .

Para cada arco  $(i, j) \in A^k, i \neq j, k \in K$ , existe una distancia (costo)  $c_{ij}^k$  y tiempo de viaje  $t_{ij}^k$ , el tiempo de servicio estará denotado como  $s_i, \forall i \in N \cup \{0\}$ , donde  $s_0 = 0$ . En cuanto a los clientes existe un intervalo de tiempo  $[e_i, l_i], \forall i \in N \cup \{0\}$ .

Sea  $d_i^k$  el tiempo de partida en el vehículo  $k$  desde cliente  $i$ ,  $\forall i \in N \cup \{0\}$  con  $d_0^k = e_0$ ,  $a_i^k$  el tiempo de llegada en el vehículo  $k$  al cliente  $i$ ,  $\forall i \in N \cup \{0\}$  con  $a_0^k = l_0$  y  $w_i^k$  el tiempo de espera del vehículo  $k$  al llegar a  $i$ ,  $\forall i \in N \cup \{0\}$ . El tiempo de llegada a un cliente  $j$  viene dado por  $a_j^k = d_i^k + t_{ij}^k$ , el tiempo de espera en  $j$  se define  $w_j^k = \max\{0, e_j - a_j^k\}$  mientras

que el tiempo de salida desde un cliente  $j$  se define  $d_j^k = a_j^k + w_j^k + s_j$ . Esta última expresión puede redefinirse como  $d_j^k = a_j^k + \max\{0, e_j - a_j^k\} + s_j$ .

Sea

$$X_{ij}^k = \begin{cases} 1, & \text{si el vehiculo } k \text{ visita al nodo } j \text{ inmediatamente despues de } i \\ 0, & \text{en cualquier otro caso} \end{cases}$$

$$\forall i, j \ 1 \leq i, j \leq |N|, i \neq j, \forall k \ 1 \leq k \leq |K|$$

El modelo matemático, que tiene asociado dos objetivos, resulta ser el siguiente:

$$\text{Minimizar } \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij}^k X_{ij}^k \quad (\text{F1})$$

Sujeto a:

$$\sum_{k \in K} \sum_{j \in N} X_{0j}^k \leq |K| \quad (\text{R1})$$

$$\sum_{k \in K} \sum_{j \in N \cup \{0\}} X_{ij}^k = 1, \ \forall i \in N \quad (\text{R2})$$

$$\sum_{j \in N} X_{0j}^k = 1, \ \forall k \in K \quad (\text{R3})$$

$$\sum_{i \in N \cup \{0\}} X_{ij}^k - \sum_{i \in N \cup \{0\}} X_{ji}^k = 0, \ \forall k \in K, \ \forall j \in V^k \setminus \{0\} \quad (\text{R4})$$

$$\sum_{i \in N} X_{i0}^k = 1, \ \forall k \in K \quad (\text{R5})$$

$$\sum_{i \in N \cup \{0\}} \sum_{j \in N \cup \{0\}} p_i X_{i,j}^k \leq P_k, \ \forall k \in K \quad (\text{R6})$$

$$\sum_{i \in N \cup \{0\}} \sum_{j \in N \cup \{0\}} v_i X_{i,j}^k \leq V_k, \ \forall k \in K \quad (\text{R7})$$

$$\sum_{k \in K} \sum_{i \in N \cup \{0\}} X_{ij}^k (a_i^k + \max\{0, e_i - a_i^k\} + s_i + t_{ij}^k) \leq l_j, \ \forall j \in N \quad (\text{R8})$$

$$\sum_{i \in N \cup \{0\}} \sum_{j \in N \cup \{0\}} X_{ij}^k (a_j^k + \max\{0, e_j - a_j^k\} + s_j + t_{j0}^k) \leq l_0, \ \forall k \in K \quad (\text{R9})$$

$$X_{ij}^k \geq 0, \ \forall k \in K, \ \forall (i, j) \in A^k \quad (\text{R10})$$

$$X_{ij}^k \in \{0, 1\}, \ \forall k \in K \forall (i, j) \in A^k \quad (\text{R11})$$

De acuerdo a lo anterior, la función F1 representa la cantidad de vehículos que saldrán del depósito y los costos asociados a cada arco, siendo el número de vehículos utilizados el objetivo primario. La restricción R1 garantiza que la cantidad de vehículos utilizados no supere al tamaño de flota ya definida. La restricción R2 impone que cada cliente sea asignado exactamente una y solo una vez a la ruta de un vehículo. Las restricciones R3-R5 garantizan que uno y solo un vehículo visite y abandone uno y solo un cliente. Las restricciones R6 y R7 garantiza la factibilidad en peso y volumen de los productos a cargar en los vehículos. La R8 señala que los vehículos no sobrepasarán la cota superior de la ventana de tiempo del cliente que deben visitar, mientras que la R9 asegura que los vehículos tienen tiempo suficiente para regresar al depósito dentro de su ventana de tiempo.

En la práctica, las restricciones R6 y R7 aplican con muy poca frecuencia al mismo tiempo, puesto que en las compañías es más común utilizar una de las dos medidas de carga sea peso (gramos, kilogramos, toneladas, etc.) o volumen (metros cúbicos, litros, galón, etc.). Resultaría poco práctico y quizás muy complejo recolectar información de dos o más características del producto, pero a fin de mantener el concepto de múltiples productos se hace necesario recolectar al menos dos características relevantes de estos.

En un ambiente real, al hablar de costos de los arcos se hace referencia a combustible utilizado, peajes, aceite, salarios, etc. Las instancias teóricas disponibles para probar el problema planteado definen estos costos en términos de la distancia que debe recorrer un vehículo para visitar los clientes asignados a este. Se puede decir que los costos reales son proporcionales a la distancia recorrida (a mayor distancia recorrida, mayor consumo de combustible por ejemplo), pero no existe una relación entre estas dos variables. No obstante, la distancia total recorrida es una aproximación apropiada para la definición de costos de viaje en cada arco y de esta manera es que son formulados los problemas de ruteo de vehículos.

En este modelo no se plantea el aspecto de la acomodación de la carga dentro del vehículo, algo que resulta ser bastante relevante en los diferentes escenarios reales del VRP. Tampoco la asignación de vehículos teniendo en cuenta la maximización de la utilización de la capacidad que en la vida real tiende a repercutir en los costos totales del enrutamiento.

### **1.3. JUSTIFICACIÓN.**

La búsqueda de una eficiente SCM conlleva a que los tomadores de decisiones estén obligados a buscar las mejores estrategias para mejorar y mantener en niveles óptimos de operación, las diferentes actividades que se desarrollan dentro de la SC, y con anterioridad se dejó claro que una de las actividades que más impacto tiene en los costos dentro de la cadena es el transporte.

Los VRP y sus variantes son escenarios que se presentan en las diferentes organizaciones que necesitan realizar la distribución de sus productos, los cuales tienen como objetivo en común el mínimo costo de operación. En nuestro entorno, las variantes tales como VRPTW, HFVRP y MPVRP son muy comunes, y los encargados de la toma de decisiones en las empresas deben dar soluciones rápidas y eficaces a dichos problemas de transporte que tiene que enfrentar, valiéndose de diferentes métodos ingenieriles ya desarrollados y su propia experiencia.

Esta investigación realiza su aporte en los siguientes aspectos:

- Nueva alternativa de solución al HFMPVRPTW: si bien es cierto que se han desarrollado muchas metodologías para resolver VRP y variantes tales como el VRPTW, HFVRP y MPVRP, es muy poca la literatura encontrada que trata de resolver la mezcla de esta terna. Hasta el momento solo se ha desarrollado un trabajo y por lo que se hace necesario dar otra alternativa distinta a fin que los tomadores de decisiones puedan dar solución al HFMPVRPTW, que resulta ser una variante más acorde con la realidad del transporte actual.

- Eficiente SCM: la idea de toda organización es aumentar sus ganancias, por lo que deben realizar todas sus operaciones al menor costo posible. Siendo el transporte una de las actividades que tiene mayor impacto en los costos finales del producto, se justifica desarrollar esta de una manera optima que permita a la organización ser competitiva en el mercado. Lo anterior implica reducir costos directamente relacionados con el transporte (combustible usado, distancia o tiempo empleado, peajes, etc.), evitar penalizaciones (por entregas tardías o no servicio), mejor utilización de los recursos (menor cantidad de vehículos, mayor aprovechamiento de la capacidad instalada), cumplimiento de los requerimientos de los clientes (cumplimiento de la demanda tanto en cantidad como en calidad del producto y tiempos de entrega) y alta capacidad de reacción ante eventualidades.
- Implementación de nuevas metaheurísticas: con el pasar del tiempo, se van desarrollando diferentes metaheurísticas que tienen como finalidad dar mejores resultados y en tiempos computacionales más adecuados que las que actualmente se tienen. Entre esas novedosas metaheurísticas se encuentra la desarrollada por Gómez Vizcaíno (2009), que es relativamente nueva y que hasta el momento no se ha explotado para la solución de otros tipos de problemas multiobjetivo, por tal razón esta es la que se utilizará para el desarrollo de esta investigación.



## **2. OBJETIVOS DE LA INVESTIGACIÓN.**

### **2.1. OBJETIVO GENERAL.**

Diseñar y desarrollar una metodología de optimización para resolver problemas de enrutamiento de vehículos caracterizados por la existencia de ventanas de tiempo para la interacción con el cliente, una flota de vehículos con capacidad restringida heterogénea y una mezcla de productos medidos en términos de peso y volumen, mediante una metaheurística basada en el fototaxismo de las bacterias que logre obtener soluciones satisfactorias y de buena calidad que sirva de base para el alcance de una eficiente SCM de cualquier organización.

### **2.2. OBJETIVOS ESPECÍFICOS.**

- Desarrollar un algoritmo que suministre soluciones satisfactorias del HFMPVRPTW teniendo en cuenta los conceptos de la metaheurística multiobjetivo basada en el fototaxismo de las bacterias al igual que otros conceptos relacionados con el VRP y sus variantes, que permitan obtener un diseño definitivo de la metodología de optimización.
- Evaluar y comparar el desempeño del algoritmo implementado con otros que dan solución al HFMPVRPTW o similares, a través de la aplicación directa sobre instancias ya conocidas, a fin de determinar que la calidad de las soluciones, obtenidas en términos de cercanía con los mejores resultados registrados, no sea en promedio superior al 15%.
- Documentar los resultados obtenidos, favorables o no, con el algoritmo computacional, a fin de identificar las ventajas y desventajas que este presenta y a su vez definir directrices para proyectos de investigación a futuro encaminados hacia mejoramiento de la metodología, diseño de nuevas técnicas de solución de

este tipo de problemas y la validación en otros entornos.

### **3. ALCANCES Y LIMITACIONES DE LA INVESTIGACIÓN.**

Esta investigación está encaminada hacia la solución del problema HFMPVRPTW, el cual ya fue definido con anterioridad, a través de un algoritmo basado principalmente en una metaheurística multobjetivo que recrea el fototaxismo de las bacterias. Las variantes menos complejas tales como el VRP, VRPTW, HFVRP y MPVRP se pueden resolver con esta metodología siempre y cuando las instancias se adapten a la diseñada para el problema a estudiar. Si se desea resolver el VRPTW, la capacidad de todos los vehículos debe ser igual y los valores de peso y volumen tanto para los productos como para la capacidad de los vehículos deben ser iguales. Por consiguiente, esta investigación también aplica en la solución de dichas variantes.

En cuanto a la aplicación, es necesario evaluar la calidad y el desempeño de esta metodología de solución, y para esto se aplicará esta en dos instancias conocidas y que fueron modificadas para que se adapten al problema a tratar en esta investigación, de las cuales ya se conocen los valores de las mejores soluciones reportadas hasta el momento. No se aplica esta metodología en una instancia real desconocida, debido a que resultaría dispendioso hacer la recolección de los datos de entrada y no se tendría una solución base con la cual hacer comparaciones en cuanto a calidad de la solución encontrada. Lo anterior refleja que la aplicación en una situación real no hace parte del alcance de la investigación, aunque puede ser una propuesta de investigación a futuro.

La formulación matemática del HFVRPTW da límite al alcance de esta investigación, debido a que dicha formulación resulta ser determinística y estática. El aplicar esta metodología a problemas de tipo estocástico y/o dinámico resulta ser tema de otras investigaciones.

Se aclara que esta investigación solo busca el diseño y desarrollo de una metodología de optimización, lo cual va acorde con lo planteado en el objetivo general. El desarrollo de una

interfaz para usuarios y/o una herramienta de tipo comercial de software no hacen parte de los objetivos planteados por esta investigación, lo que conlleva a definir el alcance de esta en estos términos.

## 4. MARCO TEÓRICO.

### 4.1. PROBLEMA DEL AGENTE VIAJERO

El problema del agente viajero (TSP por sus siglas en el idioma inglés), constituye la situación general y de partida para formular otros problemas combinatorios más complejos, aunque más prácticos, como el enrutamiento de vehículos y la programación de tareas dependientes del tiempo de alistamiento. El TSP estándar es un problema NP-*Hard*. (Aronson 1995)

El TSP consiste en encontrar una ruta de un vendedor que comienza desde un punto de partida, visita una serie de ciudades predeterminados y regresa a su punto de partida de tal manera que la distancia total recorrida sea la mínima y que cada ciudad es visitada exactamente una vez. (Gutin y Punnen 2004)

Una definición matemática formal del problema en cuestión se puede definir de la siguiente manera: sea  $G=(V, A)$  un grafo (dirigido o no dirigido), donde  $V$  representa el conjunto de los diferentes clientes o lugares a visitar,  $V = \{1,...,N\}$ , y  $A$  representa el conjunto de arcos que conectan estos sitios, donde para cada arco  $(i, j) \in A$  existe asociado un costo  $c_{i,j}$  y los arcos están relacionados con una matriz de costo  $C = (c_{i,j})_{N \times N}$ . Por lo tanto debe encontrarse un ciclo hamiltoniano<sup>2</sup> en  $G$  tal que la suma de los costos de los arcos del ciclo sea la mínima. Para este tipo de problemas, generalmente se asume que el grafo es completamente conexo.

Ahora bien, dependiendo de la matriz de costo  $C$ , el TSP se puede dividir en dos clases: el TSP simétrico (STSP), ( $G$  en este caso es no dirigido) donde  $c_{i,j} = c_{j,i}$  y el TSP asimétrico (ATSP), ( $G$  en este caso es dirigido) donde no necesariamente se cumple que  $c_{i,j} \neq c_{j,i}$ . Por

---

<sup>2</sup> Dado un grafo  $G=(V, A)$ , un ciclo hamiltoniano se define como un ciclo simple que contiene cada elemento de  $V$  exactamente una vez.

otro lado, se dice que si la matriz de costo  $C$  satisface la desigualdad triangular, es decir  $c_{i,k} + c_{j,k} \geq c_{i,j}, \forall i, j, k \in V$ , se habla de un TSP triangular.

Para los STSP, se han desarrollado heurísticas para la construcción de rutas y el mejoramiento de las mismas. Entre los algoritmos para la construcción de rutas, se destacan: *vecino más cerca*, heurísticas *greedy o voraces*, heurísticas de inserción y el algoritmo de Christofides. En cuanto a la mejora de rutas, las heurísticas *2-opt*, *3-opt* y *k-opt* han sido las más utilizadas, sin dejar de mencionar el algoritmo de Lin-Kernighan, el cual resulta ser un *k-opt* variable y fue desarrollado por (Helsgaun 2000). También algoritmos de *Ramificación y Acotamiento (Branch & Bound)* que pueden aplicarse tanto para STSP como ATSP.

Las herramientas metaheurísticas también han sido utilizadas para el mejoramiento de las rutas, puesto que estas resultan ser bastante eficientes para problemas de optimización combinatoria. SA (Kirkpatrick, Gelatt y Vechi 1983), TB (Fiechter 1994), ACO (Dorigo y Stutzle 2004) y GA han sido considerados para la solución del TSP, solo por mencionar algunos autores. Otras meta heurísticas basadas en los sistemas naturales y criaturas también se han aplicado para resolver el TSP, aunque no son tan conocidas como las anteriormente mencionadas, de las cuales se resaltan *optimización por enjambre de partículas* (Eberhart y Kennedy 1995), *optimización por colonia de abejas* (Teodorovic, y otros 2006), algoritmo de *gotas de agua inteligentes* (Shah-Hosseini 2007), *sistemas inmunes artificiales* (Dasgupta 1999) y algoritmos basados en electromagnetismo (Birbil y Fang 2003).

## 4.2. PROBLEMA DE ENRUTAMIENTO DE VEHICULOS

El problema de enrutamiento de vehículos (VRP por sus siglas en inglés) resulta ser una extensión del TSP. Este consiste en servir un grupo de clientes, en un periodo de tiempo dado, por una cantidad de vehículos pertenecientes a una flota para la cual, su capacidad no puede ser excedida. La idea principal es encontrar para cada uno de los vehículos utilizados

un ciclo hamiltoniano tal que la suma de los costos asociados a dicho ciclo sea la menor posible.

Para este tipo de problemas, al igual que el TSP, también se han desarrollado heurísticas para la solución de estos, las cuáles se pueden destacar: el *algoritmo de ahorros* (Clarke y Wright 1964) para la construcción de rutas, en el cual se inserta un cliente a una ruta teniendo en cuenta el ahorro generado por dicha inserción; los métodos de mejoramiento, como el *2-opt*, ya mencionados en la sesión de antecedentes también aplican en la solución de los VRP; métodos de *Agrupar Primero – Enrutar después* donde la idea principal es buscar un número de clientes que se asignaran a un vehículo en particular para luego a cada conjunto de clientes se le define la secuencia a visitarlos, se destacan los métodos de *asignación generalizada* (Fisher y Jaikumar 1981) y la *heurística de localización* de (Bramel y Simchi-Levi 1995); métodos de *Enrutar primero – Agrupar después* desarrollado por Beasley en el año de 1983 (Bramel y Simchi-Levi 1997) y *barrido o sweep* de (Guillet y Miller 1974), los cuales consisten en primero calcular la ruta que visita a todos los clientes resolviendo un TSP, para luego determinar las mejores particiones de ruta asignadas a cada uno de los vehículos cumpliendo sus restricciones pertinentes. Para el mejoramiento de rutas, las metaheurísticas, que han sido mencionadas en la sección de antecedentes, también han sido útiles y se han adaptado cada una de estas a la solución de los VRP.

#### **4.2.1. Variantes del VRP.**

De acuerdo a las características de las instancias del problema a solucionar, los VRP pueden ser determinísticos o probabilísticos y estáticos o dinámicos. Las variantes del VRP determinísticas y estáticas más conocidas se encuentran las siguientes:

- VRP con capacidades (CVRP): consiste en una serie de clientes o lugares que deben ser visitados por una flota idéntica de vehículos de capacidad limitada homogénea. Estos están inicialmente localizados en un depósito dado. El objetivo es encontrar

una serie de rutas para cada uno de los vehículos de mínima distancia recorrida. Cada vehículo parte desde el depósito, visita un subconjunto de clientes y regresa al depósito si violar la restricción de capacidad. Existe otro caso en donde cada cliente  $i$  tiene una demanda  $w_i$  y la restricción señala que la cantidad total entregada no puede exceder una cantidad  $Q$ . Este caso es llamado CVRP con demandas desiguales (UCVRP) (Bramel y Simchi-Levi 1997).

- VRP con restricción de distancia (DVRP): en este escenario, para cada ruta la restricción de capacidad es reemplazada por una restricción de longitud (o tiempo) máxima. En particular, una longitud no negativa  $t_{ij}$ , está asociada con cada arco  $(i, j) \in A$ , y la longitud total de los arcos de cada ruta no puede sobrepasar la máxima longitud de ruta  $T$ .
- VRP con ventanas de tiempo (VRPTW): consiste en diseñar un conjunto de rutas a un mínimo costo, que se originan y terminan en un depósito central, para una flota de vehículos que debe servir a cada cliente  $i$  con demandas conocidas dentro de un intervalo de tiempo  $[e_i, l_i]$  previamente establecido, el cual es llamado ventana de tiempo. Los clientes deben ser asignados exactamente una vez a los vehículos tal que la restricción de capacidad no sea violada.
- VRP con *Backhauls* (BVRP): en esta extensión del CVRP en la cual existen dos grupos de clientes, los *backhauls* que son clientes los cuales demandan una recolección y los *linehauls* que demandan una entrega. Existe una restricción de precedencia la cual señala que cuando una ruta atienda ambos tipos de clientes, debe atender primero todos los clientes *linehauls* antes de atender un cliente *backhaul* (Toth y Vigo 2002).
- VRP con recogida y entrega simultánea (VRPPD): en esta variante del VRP, cada cliente  $i$  tiene asociada dos cantidades  $d_i$  y  $p_i$  representando la demanda de bienes homogéneos a ser entregados y recogidos en el cliente  $i$ , respectivamente. Se asume que, en cada localización del cliente, la entrega se realiza antes de la recogida. Por lo tanto, la carga actual del vehículo antes de llegar a un lugar dado se define por la carga inicial menos todas las demandas ya entregadas más todas las demandas ya recogidas. (Toth y Vigo 2002).



- VRP con flota heterogénea (HFVRP): en esta variante los vehículos de la flota no son idénticos en alguna de sus características técnicas (capacidad de carga, velocidad, etc.). El trabajo realizado por (Ferlan y Michelon 1988) trata con vehículos que poseen diferente capacidades de carga, mientras que para la velocidad se destaca el trabajo de (Dondo y Cerdá 2007).
- VRP con múltiples productos (MPVRPTW): en este escenario la demanda de los clientes está basada en una mezcla de diferentes tipos de carga. Esto significa que los tipos de carga tiene diferentes unidades de medida, por ejemplo el peso y volumen, por lo que se hace necesario agregar restricciones en función de estas unidades de medida de la carga. Solo Dantzing y Ramsen (1959) han discutido brevemente esta variante pero sin mencionar técnicas para su solución.

#### 4.3. OPTIMIZACIÓN MULTIOBJETIVO.

Según Donoso y Fabregat (2007), un problema de optimización multiobjetivo se denota de la siguiente manera:

Optimizar [minimizar/maximizar]

$$F(X) = \{f_1(X), f_2(X), \dots, f_n(X)\}$$

Sujeto a

$$H(X) = 0$$

$$G(X) \leq 0$$

El conjunto  $F(X)$  contiene las funciones que se desean maximizar y/o minimizar, el vector  $X$  es el conjunto de variables independientes, y  $H(X)$  y  $G(X)$  representan las restricciones del modelo. Las soluciones que se encuentren resolverán el problema multiobjetivo, aun cuando estas entren en conflicto, es decir, el mejorar una puede conllevar a empeorar las otras. En este tipo de problemas se pueden presentar situaciones donde se desea maximizar o minimizar todas las funciones, y maximizar (minimizar) algunas y minimizar (maximizar) otras.

#### 4.4. OPTIMIZACIÓN COMBINATORIA.

Un problema de optimización combinatoria  $P = (S, f)$  es un problema de optimización en el cual dado un conjunto finito de las soluciones factibles  $S$  y una función objetivo  $f: S \rightarrow \mathbb{R}$  que asigna un valor a cada uno de los elementos de  $s \in S$ . La idea es encontrar el mejor elemento, en términos de maximización o minimización según sea el caso, del conjunto  $S$ . Estos elementos típicamente son números enteros, subconjuntos de un conjunto de elementos, permutaciones de un conjunto de elementos o grafos.

Los problemas de optimización pueden ser resueltos de manera exacta encontrando una solución óptima lo que implica realizar una revisión completa a todo el espacio solución de este, pero cuando dicho espacio resulta ser muy amplio la búsqueda del óptimo llevaría mucho tiempo y esfuerzo. Según Garey y Johnson (1979), los *algoritmos* son procedimientos paso a paso que sirven para resolver problemas, y se dice que este es el “*más eficiente*” si resulta ser el más rápido (desde el punto de vista de complejidad computacional). Para los problemas TSP y VRP, junto con sus variantes, hasta el momento no se ha podido encontrar un algoritmo que sea capaz de resolverlos en tiempo polinomial, por lo cual dichos problemas son del tipo *NP* y, en el peor de los casos, los algoritmos que intentan resolver este tipo de problemas emplearán un tiempo exponencial del tamaño de la entrada del problema, donde en los TSP y VRP aumenta a medida que existan más clientes a visitar. Para una mayor profundización de la temática de *complejidad computacional* el lector puede consultar a Garey y Johnson (1979).

Una mejor alternativa a los métodos exactos resulta ser los métodos aproximativos, en los cuales no se garantiza encontrar soluciones óptimas, sino soluciones muy cercanas a esta en una cantidad de tiempo significativamente pequeño. Entre los métodos aproximativos más básicos se encuentran las *heurísticas de construcción* y *métodos de búsqueda local*.

##### 4.4.1. Heurísticas de Construcción.

De acuerdo con Alba (2005) las *heurísticas de construcción*, que típicamente resultan ser las más rápidas, generan soluciones desde el principio mediante la adición de componentes

de solución definidos a una solución parcial vacía inicialmente. Lo anterior se realiza hasta que se encuentre una solución factible o se satisfaga un criterio de parada. Las soluciones (sean parciales o no) son secuencias  $\{c_i, \dots, c_k\}$  compuestas de componentes de solución  $c_j$  de conjunto finito de componentes de solución  $C$  (para los problemas de ruteo de vehículos este conjunto está compuesto por los clientes a visitar). Llámese  $s$  a una solución completa y  $s^p$  a una solución parcial.

Dichas heurísticas deben definir *a priori* el conjunto de posibles extensiones para cada solución (parcial) factible  $s^p$  y dicho conjunto se denota como  $N$ , que es un subconjunto de  $C \setminus \{c | c \in s^p\}$ . En cada momento de la construcción una de las posibles extensiones se escoge hasta que  $N = \emptyset$ , lo que significa que  $s^p$  es una solución (sea completa o parcial) que no puede ser extendida a una solución factible. El algoritmo de construcción (Alba 2005) se define así:

$s^p = \{\}$

Determinar  $N$

**Mientras que  $N \neq \emptyset$  haga**

$c \leftarrow$  un elemento del conjunto  $N$

Al conjunto  $s^p$  añada  $c$ .

Actualizar  $N$

**Fin Mientras que**

Imprima  $s^p$

Entre las diferentes heurísticas de construcción se encuentran los métodos de ahorro, formación de clústeres y técnicas de inserción que ya fueron mencionadas con anterioridad.

#### 4.4.2. Métodos de Búsqueda Local.

Los algoritmos de búsqueda local parten de una solución inicial e iterativamente tratan reemplazarla por una mejor en una vecindad definida de la actual solución. Una vecindad de una solución es un conjunto de soluciones que se derivan de otra, mediante una regla o

función llamada estructura de vecindad. Esta es una función  $f: S \rightarrow 2^S$  que asigna a cada solución  $s \in S$  un conjunto de vecinos  $N(s) \subseteq S$ .  $N(s)$  es la vecindad de  $s$ . Los algoritmos de búsqueda local se definen de la siguiente forma.

$s \leftarrow$  Una solución inicial de  $S$

**Mientras que**  $\exists s' \in N(s)$  tal que  $f(s')$  es mejor que  $f(s)$  **haga**

$s \leftarrow s'$

**Fin Mientras que**

Imprima  $s$

Entre los métodos de búsqueda local se encuentran los diferentes algoritmos de mejora de rutas de los que ya hablaron con anterioridad.

#### 4.4.3. Metaheurísticas.

Al hablar de metaheurísticas se hace referencia a estrategias algorítmicas de alto nivel las cuales se usan para guiar otras heurísticas o algoritmos en la búsqueda de espacios de soluciones factibles del valor óptimo (un solo objetivo) o conjunto de valores óptimos (múltiples objetivos) (Donoso y Fabregat 2007). Las principales características de las metaheurísticas son (Alba 2005):

- Tienen como meta explorar de manera eficiente el espacio de búsqueda con el fin de encontrar soluciones cercanas al óptimo.
- Van desde simples procedimientos de búsqueda local a procesos de aprendizaje complejos.
- Son aproximativos y usualmente no determinísticos.
- Incorporan mecanismos para evitar quedar atrapados en óptimos locales.
- Los conceptos básicos de metaheurísticas no están atados a un problema en específico y se pueden adaptar a cualquier problema.
- Pueden usar algún tipo de memoria para guiar la búsqueda.

Existen dos aspectos de gran importancia al usar una metaheurística. El primero de ellos es la *diversificación* que hace referencia a la exploración del espacio de búsqueda mientras que el segundo aspecto llamado *intensificación* está relacionado con la explotación de la experiencia acumulada de búsqueda. El balance entre estos dos elementos es vital, por un lado para identificar rápidamente regiones en el espacio de búsqueda con soluciones de alta calidad y por otro lado no gastar mucho tiempo en explorar espacios de solución que ya hayan sido explorados o que no posean soluciones de alta calidad (Alba 2005).

Entre las metaheurísticas más conocidas en la actualidad se encuentran SA, ACO, TS y GA, pero también hay que destacar otras como las basadas en *enjambres de partículas*, *colonia de abejas*, *gotas de agua inteligentes*, *sistemas inmunes artificiales* y *electromagnetismo*.

#### **4.5. APRENDIZAJE DE MAQUINA Y APRENDIZAJE REFORZADO**

Es posible afirmar que el aprendizaje se presenta cuando interactuamos con nuestro ambiente. Un niño juega, salta, corre u observa a su alrededor y para tales actividades no existe una persona que le indique como hacerlas, simplemente tiene una conexión sensorial y motora directa con el medio ambiente. Durante estas interacciones con el medio ambiente se va adquiriendo gran cantidad de información acerca de las consecuencias que generan las acciones y que se debe hacer para alcanzar ciertos objetivos, y con esta cantidad de información recolectada se logra tomar las decisiones más convenientes para resolver un problema. El elemento que enfrenta el problema, aprende de su interacción con el *medio ambiente* y toma decisiones es llamado *agente*.

El *aprendizaje de máquina*, *LM* por sus siglas en inglés, es un área de la inteligencia artificial que busca resolver problemas teniendo en cuenta el aprendizaje obtenido a través de la experiencia. Este consiste en desarrollar algoritmos para optimizar un criterio de desempeño usando datos de referencia o experiencia pasada, donde se tiene un modelo con ciertos parámetros y el aprendizaje es la ejecución del programa para optimizar parámetros del modelo usando la experiencia pasada. Dicho modelo puede ser predictivo para hacer predicciones en el futuro, o descriptivo para ganar conocimiento de los datos, o ambos

(Alpaydin 2010). Existen tres tipos de *LM*: el *aprendizaje supervisado* consiste en comparar los resultados obtenidos al finalizar el programa contra un valor de correcto u óptimo, mientras que en el *aprendizaje no supervisado* no se conocen los valores correctos u óptimos y es necesario analizar, a partir de los datos de entrada, la ocurrencia de ciertos patrones y observar que ocurre y que no.

Una combinación entre el *aprendizaje supervisado* y el *no supervisado* origina el tercer tipo de aprendizaje: *aprendizaje reforzado*. En este tipo de aprendizaje, el *agente* debe tomar decisiones durante su interacción con el *medio*, el cual responde durante esta interacción y presenta nuevas situaciones al agente. Sin embargo, el *medio* también le da una recompensa al *agente* como consecuencia de sus actos, por lo tanto es responsabilidad del *agente* definir las acciones que le generen una mayor recompensa por parte del *medio*. En tal caso, el algoritmo de aprendizaje de maquina debe evaluar las bondades de cada acción y aprender de las acciones realizadas en el pasado y así generar una *política* que le genere el mayor beneficio. Tal método de aprendizaje se conoce como *algoritmo de aprendizaje reforzado* (Alpaydin 2010) (Sutton y Barto 1998).

#### **4.6. HEURISTICA DE OPTIMIZACIÓN MULTI OBJETIVO BASADA EN EL FOTOTAXISMO DE LAS BACTERIAS.**

La estrategia multiobjetivo creada por Gómez Vizcaíno (2009) recrea el comportamiento de una colonia de bacterias y su fototaxis, es decir, el proceso en que incurre esta colonia, caracterizada por utilizar la luz como fuente de energía, desde su nacimiento, pasando por la replicación, mutación y fotosíntesis, hasta llegar a la muerte. Los diferentes pasos que sigue este algoritmo son los siguientes:

**Colonia\_Bacterias\_Iniciales:** Este procedimiento consiste en la generación de la colonia inicial de bacterias, es decir el conjunto de soluciones factibles iniciales del problema. Dichas soluciones se generan a criterio del programador de acuerdo con el problema a solucionar a las cuales se les debe asignar una cantidad de energía inicial.

**Ciclos Bacterianos:** hace referencia a la condición de parada del ciclo repetitivo al que se verán sometidas las soluciones iniciales del problema.

**Giro\_Bacterias:** las bacterias que tengan energía suficiente para realizar un giro pueden aplicar una “voltereta” en la que se realiza la búsqueda de las diferentes direcciones en las cuales la bacteria puede moverse hacia la fuente de luz. El programador define estas direcciones, que representan los diferentes cambios que puede presentar el conjunto de soluciones (colonia de bacterias) hacia otras soluciones, y dicho proceso implica una pérdida de energía para cada solución, la cual es definida por el programador. Estas direcciones pueden ser evaluadas mediante la siguiente fórmula:

$$GLS_i = \frac{\prod_{r=1}^{\frac{nobj}{2}} \exp\left(\frac{-f_{2*r-1}(B_i)}{\max(f_{2*r-1}(C))}\right)}{\prod_{r=1}^{\frac{nobj}{2}} \exp\left(\frac{f_{2*r}(B_i)}{\max(f_{2*r}(C))}\right)}; B_i \in C \quad (1)$$

En la expresión anterior,  $B_i$  es la dirección de la bacteria que hace parte del conjunto de direcciones  $C$  en las que esta se puede mover. La cantidad de funciones objetivo que se busca optimizar se denota  $nobj$  y  $f_r$  es la función objetivo  $r$ . En el Anexo 1 se encuentra una explicación más detallada para el uso de esta fórmula.

**Carrera\_hacia\_la\_luz:** una vez obtenido el conjunto de direcciones de cada bacteria, esta debe seleccionar aleatoriamente entre las cuatro mejores que le permitirá estar más cerca de la fuente de luz. Esto significa que la estructura de la solución cambiará, con tendencia hacia los objetivos del problema que se desea solucionar. Luego de seleccionada la dirección, la bacteria hace una carrera hacia la fuente de luz con un rango definido por el usuario. La bacteria debe tener energía suficiente para desplazarse y una vez realizado dicho desplazamiento, esta perderá una cierta cantidad de energía definida por el programador.

**Fisión Binaria:** posterior al desplazamiento, la bacteria se divide y la nueva bacteria que nace posee una cantidad de energía inicial. Lo anterior ocurre si dicha bacteria o solución tiene energía suficiente, para después perder cierta cantidad de energía (definida por el programador) una vez realizada la división.

**Mutación\_espontánea:** la bacteria realiza un cambio en su estructura y posición con respecto a la fuente de luz y esta puede mejorar o empeorar. Dicha mutación solo se realiza sobre una cantidad de bacterias seleccionadas por probabilidad, las cuales deben tener energía suficiente para llevar a cabo este proceso. Al igual que los anteriores procesos, este conlleva a una reducción de energía la cual es definida por el programador.

**Mutación\_reversa:** este proceso solo se lleva a cabo en un porcentaje de las bacterias que realizaron la mutación espontánea, las cuales son seleccionadas por probabilidad, y que tienen energía suficiente para llevar a cabo la mutación reversa. Si estas bacterias desmejoraron con respecto a la luz, deben regresar a la estructura que tenían con anterioridad, es decir, deben aplicar mutación reversa. Dicho proceso implica una pérdida de energía la cual es definida por el programador.

**Selección\_muerte\_bacterias:** en este punto, las bacterias que no tengan energía o sea muy baja, dejarán de pertenecer a la colonia puesto que se alejaron mucho de la luz y no tienen energía suficiente para realizar un giro y desplazarse hacia una posición más cercana de la fuente de luz. Una vez eliminadas estas bacterias, las restantes se unen junto con aquellas que nacieron del proceso de fisión binaria.

**Fotosíntesis:** la luz es el alimento de las bacterias y la transformación de esta en energía es la que le permite sobrevivir. La cantidad de energía que gana la bacteria mantiene una relación directamente proporcional a la distancia que se encuentra esta con la fuente de luz. Esta cantidad de energía obtenida por cada bacteria  $i$  se explica mediante la siguiente expresión:

$$ATP_i = energia\_por\_fotosintesis * GLS_i, B_i \in C \quad (2)$$



La variable *energía\_por\_fotosíntesis* es definida por el programador. En la expresión anterior,  $B_i$  es la bacteria o solución que hace parte de la colonia  $C$ .  $nobj$  es la cantidad de funciones objetivo que se busca optimizar y  $f_r$  la función objetivo.

**Selección\_de\_frente\_pareto:** una vez culminados los ciclos bacterianos, la colonia resultante contiene bacterias (soluciones) dominadas y no dominadas, por lo que es necesario escoger las bacterias no dominadas y con estas se define la colonia final. Los conceptos de dominancia y no dominancia entre soluciones, pueden revisarse en Donoso y Fabregat (2007).

## 5. PROPUESTA METAHEURISTICA PARA EL MPHVRPTW

La alternativa de solución propuesta para el MPHVRPTW consiste en un algoritmo basado principalmente en el fototaxismo de las bacterias, pero con la adición de una rutina enfocada hacia la minimización del número de vehículos utilizados, la cual se caracteriza por permitir inserciones no factibles de manera temporal seguida de intentos por restaurar su factibilidad. Luego el algoritmo bacterial se encargará de realizar esfuerzos por minimizar el segundo objetivo, la distancia total recorrida, mediante técnicas de búsqueda local como es la *inserción*, y que será explicada en detalle más adelante.

La meta heurística consiste en lo siguiente:

### Procedimiento Metaheuristica\_MPHVRPTW

#### Inicio

Colonia\_Bacterias\_Iniciales ( )

Reducción\_rutas ( )

*CB := Ciclos Bacterianos*

**Para** *cb = 1* hasta *CB*

Giro\_Bacterias ( )

Carrera\_hacia\_la\_luz ( )

Fisión\_binaria ( )

Mutación\_Espontánea ( )

Mutación\_Reversa ( )

Selección\_muerte\_bacterias ( )

Fotosíntesis ( )

**Fin Para**

Selección\_mejor\_solución ( )

**Fin**

## 5.1. DESCRIPCIÓN DE LA METAHEURISTICA.

### 5.1.1. Colonia de bacterias iniciales.

Esta etapa consta de la construcción de las rutas iniciales del problema, es decir el nacimiento de las bacterias (soluciones) iniciales. Sea  $SI$  la cantidad de soluciones iniciales donde  $SI \in \mathbb{Z}^+$ ,  $S_s$  la solución creada y  $C$  la colonia de soluciones donde  $S_s \in C$ . Sea  $N = \{1, 2, \dots, n\}$  el conjunto de clientes no visitados,  $K$  el conjunto de vehículos disponibles para ser programados donde  $1 \leq k \leq |K|$ ,  $R_k$  la secuencia de clientes a visitar por el vehículo  $k$  por lo tanto  $S_s = \{R_1, R_2, \dots, R_k\}$ ,  $N_f$  el conjunto de clientes factibles (que cumplen con las restricciones de peso, volumen, ventana de tiempo del cliente y tiempo de regreso al punto inicial) a visitar donde  $N_f \subseteq N$ ,  $i_m$  el cliente en el cual se encuentra el vehículo y  $i_{m+1}$  el siguiente cliente a visitar tal que  $i_{m+1} \in N_f$  por consiguiente  $R_k = \{i_0, i_1, i_2, \dots, i_m\}$ , donde  $i_0 = i_m = 0$  y representa el punto de partida del vehículo  $k$ ;  $h$  el cliente factible de insertar entre  $i_m$  y  $i_{m+1}$ , tal que  $h \in N$ , mientras que  $c_{i_m i_{m+1}}$  es la distancia o tiempo de viaje que se emplea para ir de  $i_m$  hacia  $i_{m+1}$ , y existe un intervalo de tiempo  $[e_{i_{m+1}}, l_{i_{m+1}}], \forall i_{m+1} \in N_f$ . Cada ruta  $R_k \in S_s$  tiene una distancia recorrida  $TDR_k = \sum_{m=0}^{|R_k|-1} c_{i_m i_{m+1}}, \forall i_m \in R_k$  y la distancia total recorrida de la solución es  $TTDS_s = \sum_{k=1}^{|S_s|} TDR_k$ . El usuario debe asignar a cada solución una energía inicial  $ES_s$ .

Este procedimiento consiste en seleccionar el siguiente cliente a visitar teniendo en cuenta la mínima distancia empleada para llegar a este y a su vez el menor  $l_{i_{m+1}}$  (tiempo límite para ser visitado). Una vez construida la ruta, se intenta añadir, de manera aleatoria, la mayor cantidad de clientes posibles no visitados a esta que generen la menor distancia empleada y que tengan menor tiempo límite a ser visitado. Lo anterior se lleva a cabo utilizando la formula  $GLS_i$  mostrada con anterioridad. Dicho procedimiento puede observarse a continuación:

## Procedimiento Colonia\_Bacterias\_Iniciales ( )

### Inicio

Para  $s = 1$  hasta SI

$$i_m = 0, k = 1, R_k = \{0\}, S_s = \emptyset, ES_s = E$$

Determine  $N$  como el conjunto de clientes no visitados

Mientras que  $N \neq \emptyset$  haga

Determinar  $N_f$  como el conjunto de clientes factibles a visitar

Si  $N_f = \emptyset$  entonces

$$i_m = 0$$

Añada  $i_m$  a  $R_k$

Mientras que  $\exists h \in N$  haga

Seleccione al azar un arco  $(i_m, i_{m+1}) \in R_k$

Escoja el cliente  $h$  factible a insertar en dicha posición de acuerdo a

$$h = ArgMax \left\{ \frac{\exp\left(\frac{-c_{imh}}{\max\{c_{imh}\}}\right)}{\exp\left(\frac{l_h}{\max\{l_{i_h}\}}\right)} \right\}, \forall h \in N$$

Añada  $h$  a  $R_k$  y elimine  $h$  del conjunto  $N$

Fin Mientras que

Añada  $R_k$  a  $S_s$

$$k = k + 1$$

$$R_k = \{0\}$$

Si no

$$i_{m+1} = ArgMax \left\{ \frac{\exp\left(\frac{-c_{im+1}}{\max\{c_{im+1}\}}\right)}{\exp\left(\frac{l_{im+1}}{\max\{l_{im+1}\}}\right)} \right\} \forall i_{m+1} \in N_f$$

$$i_m = i_{m+1}$$

Añada  $i_m$  a  $R_k$  y remueva a  $i_{m+1}$  del conjunto  $N$

Fin Si

Fin Mientras que

Añada  $S_s$  a  $C$

**Fin Para**

**Fin**

Cabe aclarar que la asignación de vehículos se realiza en orden descendente de acuerdo a su capacidad, con el fin de que esta permita reducir de manera significativa la cantidad de rutas a emplear.

### 5.1.2. Reducción del número de rutas empleadas.

En este punto se atacará uno de los primeros objetivos del problema el cual consiste en minimizar la cantidad de rutas empleadas y para alcanzar este fin se debe eliminar la ruta más pequeña de esta solución denotada por  $SRS_s = ArgMin\{|R_k| \}, \forall R_k \in S_s$ . La idea principal radica en insertar todos los clientes de dicha ruta en cualquier posición de las restantes. Sin embargo, en algunas ocasiones esta ruta no puede ser eliminada en su totalidad y es aquí donde se toman las ideas desarrolladas por algunos autores donde se realizan inserciones no factibles para luego realizar movimientos de clientes tendientes a regresar la factibilidad de la solución (Nagata y Bräysy 2009).

Sea  $NFS_s$  la vecindad de soluciones no factibles generadas por insertar un cliente  $h \in SRS_s$  en diferentes arcos  $(i_m, i_{m+1})$  de las restantes rutas diferentes a la que dicho cliente pertenece. Estas no factibilidades consisten en el no cumplimiento de las restricciones del problema por parte de los clientes  $\{i_{m+1}, i_{m+2}, \dots, i_0\}$ , el cliente  $h$  si las debe cumplir. Estas soluciones pueden ser evaluadas mediante una función de penalidad en términos de violación de capacidad y violación de las ventanas de tiempo (Nagata y Bräysy 2009). La violación en términos de capacidad  $P_Q(IS_s)$  se define como la suma del excedente total de las demandas en todas las rutas de la solución, que para el problema tratado en esta investigación será el excedente total de las demandas en términos de peso y volumen.

La violación en ventanas de tiempo se define de la siguiente manera: dada una ruta  $R_k$ , la restricción de ventana de tiempo se viola si el tiempo de llegada a un cliente  $a_{i_m}$  resulta ser  $a_{i_m} > l_{i_m}$  (recuerde la formulación del problema en la sección 1.2). Para este caso se asume que el vehículo podría regresar y cumplir con la cota de tiempo  $l_{i_m}$ , si se paga una penalidad de  $(a_{i_m} - l_{i_m})$ . Al insertar este cliente no factible, esta penalización se puede pagar más de una vez en la ruta en la cual se insertó el cliente, por tanto la penalización total por ventana de tiempo en la solución  $P_{TW}(IS_s)$  es la suma de todas las penalizaciones cometidas en la solución no factible.

Entre todos los elementos de  $NFS_s$  se debe buscar aquella solución que tenga la menor penalización posible usando la siguiente expresión:

$$mejor\_IS_s = ArgMin\{F(IS_s) = P_Q(IS_s) + P_{TW}(IS_s)\} \quad (3)$$

Se busca encontrar la menor penalización ya que garantiza obtener la menor cantidad de clientes no factibles, visitados después del cliente  $h$ , que deban ser “expulsados” de la ruta en la cual este se insertó, y este conjunto de clientes expulsados está definido como  $EC$ . Los clientes pertenecientes a  $EC$  se intentan insertar en el resto de rutas diferentes a  $SRS_s$ , en el orden que fueron expulsados y en cualquier posición, dichas inserciones se realizan con el fin de que  $EC = \{\emptyset\}$ . Si esto no sucede, los clientes del conjunto  $EC$  se añaden a  $SRS_s$  y se construyen rutas de acuerdo a lo expresado en la rutina Colonia\_Bacterias\_Iniciales( ).

El procedimiento de reducción de rutas se lleva a cabo durante un número determinado de veces  $RC$ , siendo  $RC \in \mathbb{Z}^+$ , y se describe a continuación:

#### **Procedimiento Reducción\_rutas ( )**

##### **Inicio**

**Para**  $s = 1$  **hasta**  $|C|$

**Para**  $rc = 1$  **hasta**  $RC$

$$S''_s = S_s$$

Determinar  $SRS_s = ArgMin\{|R_k|\}, \forall R_k \in S_s$

**Mientras que**  $\exists i_m \in SRS_s$  factibles de inserción en toda  $R_k \in S_s \setminus \{SRS_s\}$  **haga**

    Inserte cada  $i_m \in SRS_s$  en cualquier lugar de cualquier  $R_k \in S_s \setminus \{SRS_s\}$

**Fin mientras que**

**Si**  $|S_s| < |S''_s|$  **entonces**

    Actualice  $S_s$

**Si no**

**Mientras que**  $\exists h \in SRS_s$  **haga**

        Genere  $NFS_s, \forall h \in SRS_s$

        Determine  $mejor\_IS_s = ArgMin\{F(IS_s) = P_Q(IS_s) + P_{TW}(IS_s)\}$

$S_s = mejor\_IS_s$

        Determine el conjunto de clientes  $EC$

        Actualice  $S_s$

        Determine los clientes  $i_m \in EC$  factibles de inserción

        en toda  $R_k \in S_s \setminus \{SRS_s\}$

        Inserte cada  $i_m \in EC$  en cualquier lugar de cualquier  $R_k \in S_s \setminus \{SRS_s\}$

        en el orden que fueron expulsados

**Fin mientras que**

**Si**  $EC = \{\emptyset\}$  **entonces**

        Actualice  $S_s$

**Si no**

        Genere rutas para  $EC$  usando rutina Colonia\_Bacterias\_Iniciales( )

        Actualice  $S_s$

**Si**  $|S_s| > |S''_s|$  **entonces**

$S_s = S''_s$

**Fin Si**

**Fin Si**

**Fin Si**

**Fin Para**

**Fin Para**

Determinar la mejor solución y eliminar del conjunto  $C$  aquellas con los peores resultados

**Fin**

### 5.1.3. Ciclos Bacterianos.

Como se mencionó con anterioridad, este consiste en la condición de parada del ciclo repetitivo al que se verán sometidas todas las soluciones generadas en la rutina `Colonia_Bacterias_Iniciales()`. La cantidad de ciclos bacterianos  $CB$  está definida por el usuario.

Antes de dar inicio al ciclo, se genera un conjunto llamado  $SF$  que resulta ser copia de las soluciones que se encuentran en la colonia  $C$  y cuyos elementos estarán denotados como  $SFS_s$ .

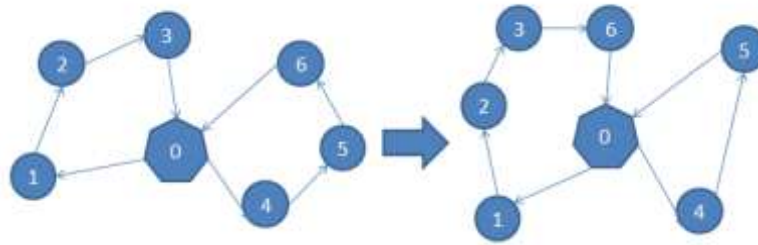
### 5.1.4. Giro Bacterias.

En este instante, cada bacteria o solución  $S_s$  debe realizar un giro y buscar todas las posibles direcciones en las cuales puede desplazarse. Esto significa la generación de la vecindad de cada solución mediante métodos de búsqueda local, y dicho método será el siguiente:

- *Insertión:* consiste en seleccionar un cliente  $h$  de una ruta  $R_k$  e insertarlo en todas las posiciones factibles de todas las rutas que pertenecen a la solución  $S_s$ . En la
- Figura 1 se puede observar un ejemplo de dicho método.

Figura 1. Inserción de un cliente.





Fuente: Elaboración propia.

El método de búsqueda local se realizara a un número de clientes escogidos al azar, variable definida como *radio\_de\_giro*, y así obtener la vecindad generada por cada uno de estos. Una vez obtenida sus vecindades se debe escoger por cada cliente la solución con la menor distancia total recorrida es decir la menor  $TTDS_s$  y añadirla al conjunto *Mejores $S_s$* . El proceso anterior solo se realiza a las soluciones que tengan la energía suficiente y una vez finalizado, dichas soluciones pierden una cantidad de energía llamada *desgaste\_por\_giro*. Dicha rutina se describe a continuación:

#### Procedimiento Giro\_Bacterias ( )

##### Inicio

Para  $s = 1$  hasta  $|C|$

Si  $ES_s > \text{desgaste\_por\_giro}$  entonces

Generar vecindad de *radio\_de\_giro* clientes escogidos al azar mediante rutina de inserción

$\forall i_m \in S_s$

Obtenga la solución con menor  $TTDS_s$  para cada uno de los clientes

Generar *Mejores $S_s$*

$ES_s = ES_s - \text{desgaste\_por\_giro}$

Fin Si

Fin Para

Fin

#### 5.1.5. Carrera hacia la luz.

Una vez las bacterias hayan realizado el giro, las soluciones perderán una cantidad de energía *desgaste\_por\_carrera*, definida por el usuario, al tener que desplazarse hacia esta nueva posición siempre y cuando tenga la energía suficiente para este desplazamiento. La posición a la cual esta se desplazara será la mejor solución del conjunto *Mejores\_S<sub>s</sub>*, conjunto proveniente de la rutina anterior.

Como paso adicional, si después de haber hecho la rutina de Giro\_Bacterias ( ) no se logra encontrar una mejora en distancia con respecto a *SFS* durante los últimos *n*-ciclos bacterianos, variable definida como *largo\_carrera*, dicha solución se retira de la colonia *C*.

**Procedimiento** Carrera\_hacia\_la\_luz( )

**Inicio**

**Para** *s* = 1 hasta *|C|*

**Si** *ES<sub>s</sub>* > *desgaste\_por\_carrera* **entonces**

    Seleccionar del conjunto *Mejores\_S<sub>s</sub>* aquella con la menor *TTDS<sub>s</sub>*

    Actualizar *S<sub>s</sub>*

*ES<sub>s</sub>* = *ES<sub>s</sub>* - *desgaste\_por\_carrera*

*Mejores\_S<sub>s</sub>* = { $\emptyset$ }

**Fin Si**

**Si** *TTDS<sub>s</sub>* ≥ *TTDSFS* durante los últimos *largo\_carrera* ciclos bacterianos **entonces**

    Elimine *S<sub>s</sub>* de la colonia *C*

**Si no**

    Actualice *SFS* a la mejor solución encontrada de la colonia *C*

**Fin Si**

**Fin Para**

**Fin**

#### 5.1.6. Fisión Binaria.

En este punto aquella solución  $S_s$  que tenga energía suficiente para duplicarse lo hará para luego perder una cantidad de energía *desgaste\_por\_fisión* y dicha “copia” ingresará en un conjunto denotado  $DC$ . Por lo tanto la rutina de fisión binaria se muestra a continuación:

**Procedimiento** Fisión\_binaria ( )

**Inicio**

**Para**  $s = 1$  **hasta**  $|C|$

**Si**  $ES_s > \text{desgaste\_por\_fisión}$  **entonces**

Añada copia de  $S_s$  a  $DC$  denotada como  $S_{s+1}$

$$ES_{s+1} = E$$

$$ES_s = ES_s - \text{desgaste\_por\_fisión}$$

**Fin Si**

**Fin Para**

**Fin**

### 5.1.7. Mutación Espontánea.

Una parte de la población de soluciones de la colonia  $C$ , seleccionada por probabilidad *prob\_mut\_espon*, presentará un cambio en su estructura siempre y cuando tenga la energía suficiente para esto, para así después perder una cantidad de energía *desgaste\_mutación\_espontánea*. Dicho cambio se realizara con la rutina de *inserción* pero de tipo aleatoria, es decir, se seleccionara un cliente cualquiera y se insertara en la primera posición factible que se encuentre

El procedimiento de mutación espontánea se describe de la siguiente manera:

**Procedimiento** Mutación\_Espontánea ( )

**Inicio**

**Para**  $s = 1$  **hasta**  $|C|$

Aleatorio = Random ( )

**Si**  $(ES_s > \text{desgaste\_mutación\_espontánea}) \wedge (\text{Aleatorio} > \text{prob\_mut\_espon})$

$$S''_s = S_s$$

Aplicar rutina de inserción aleatoria en la solución  $S_s$

Actualizar  $S_s$

$$ES_s = ES_s - \text{desgaste\_mutación\_espontánea}$$

**Fin Si**

**Fin Para**

**Fin**

### 5.1.8. Mutación reversa.

Dentro grupo de soluciones que entraron al proceso de Mutación\_Espontánea ( ) se seleccionan por probabilidad  $prob\_mut\_reversa$  aquellas soluciones, con energía suficiente, que deben entrar al proceso de la mutación reversa, el cual tiene como objetivo analizar si la solución mejora o empeora (de acuerdo a la cantidad de rutas  $NRS_s$  y distancia total recorrida  $TTDS_s$ ) después de haber hecho la mutación espontánea para tomar la decisión de regresar o no al estado anterior al haber mutado. Esto último implica una pérdida de energía  $desgaste\_mutación\_reversa$  definida por el usuario. El procedimiento de mutación reversa se describe a continuación:

**Procedimiento** Mutación\_Reversa ( )

**Inicio**

**Para**  $s = 1$  **hasta**  $|C|$

Aleatorio = Random ( )

Si  $(S_s \text{ mutó}) \wedge (ES_s > \text{desgaste\_mutación\_reversa}) \wedge (\text{Aleatorio} > \text{prob\_mut\_reversa})$  entonces

**Si**  $(NRS_s \geq NRS''_s) \wedge (TTDS_s > TTDS''_s)$  entonces

$$S_s = S''_s$$

$$ES_s = ES_s - \text{desgaste\_mutación\_reversa}$$

**Fin Si**

**Fin Si**

**Fin Para**

**Fin**

### 5.1.9. Selección Muerte de Bacterias.

Aquellas soluciones que no tengan energía o sea muy baja, no podrán seguir haciendo parte de la colonia de bacterias puesto que se encuentran alejadas de luz, es decir la calidad de dichas soluciones no es buena y no tienen la energía suficiente para entrar al procedimiento Giro\_Bacterias ( ) que les permita desplazarse hacia una mejor solución. Una vez realizado este proceso, se añaden las soluciones nuevas que fueron producto del proceso de fisión binaria. Esta rutina se describe a continuación:

**Procedimiento** Selección\_muerte\_bacterias ( )

**Inicio**

**Para**  $s = 1$  **hasta**  $|C|$

**Si**  $ES_s \leq 0$  **entonces**

        Eliminar a  $S_s$  del conjunto  $C$

**Fin Si**

**Fin Para**

$C = \{C \cup DC\}$

**Fin**

### 5.1.10. Fotosíntesis.

Una vez descartadas las bacterias con poca energía, debe estudiarse entre la colonia de soluciones cuales se encuentran más cerca a luz puesto que existe una relación directamente proporcional entre la cercanía a esta y la cantidad de energía que gana la solución. La cercanía a la luz viene dada por la calidad de la solución obtenida, es decir soluciones que tengan la menor cantidad de vehículos  $NRS_s$  y menor distancia total recorrida  $TTDS_s$  serán las que ganen mayor cantidad de energía con respecto a las demás pertenecientes a la colonia  $C$ . El usuario define la variable *energía\_por\_fotosíntesis* y cada bacteria de la colonia obtendrá energía de acuerdo a la siguiente expresión:

$$ES_s = ES_s + \text{energía\_por\_fotosíntesis} * \left( \frac{\exp\left(\frac{-NRS_s}{\max\{NRS_s\}}\right)}{\exp\left(\frac{TTDS_s}{\max\{TTDS_s\}}\right)} \right), \forall S_s \in C$$

A medida que la solución gane más energía tiene más posibilidades de sobrevivir, lo que garantiza que al finalizar todo el ciclo repetitivo se obtendrán las mejores soluciones. Este procedimiento se describe de la siguiente manera:

**Procedimiento** Fotosíntesis ( )

**Inicio**

**Para**  $s = 1$  **hasta**  $|C|$

$$ES_s = ES_s + \text{energía\_por\_fotosíntesis} * \left( \frac{\exp\left(\frac{-NRS_s}{\max\{NRS_s\}}\right)}{\exp\left(\frac{TTDS_s}{\max\{TTDS_s\}}\right)} \right), \forall S_s \in C$$

**Fin Para**

**Fin**

#### 5.1.11. Selección de la mejor solución.

Una vez culminado todo el proceso del ciclo bacteriano, se seleccionara únicamente la solución que tenga la menor cantidad de rutas  $NRS_s$  y menor distancia total recorrida  $TTDS_s$ , es decir la solución  $SFS$  resultante de todo el proceso. Al ser este un problema de tipo lexicográfico, solo se hace necesario obtener una única solución.

Como paso adicional, para esta última solución encontrada es necesario realizar nuevamente una reasignación vehicular con el fin de obtener una mejor utilización de la capacidad de cada vehículo, ya que dicha asignación se hizo por orden descendente de acuerdo a su capacidad en peso y volumen. Para esto se asigna a cada ruta el vehículo con la menor capacidad posible que logre transportar la carga en dicha ruta.

## 6. EXPERIMENTACION Y RESULTADOS.

Para evaluar la propuesta metaheurística, se dará solución a dos instancias creadas por Marius Solomon, identificadas como R201 y R209, las cuales ya fueron modificadas para que se adapten a la variante a solucionar en esta investigación. Originalmente las instancias propuestas por Solomon se caracterizan por tener la capacidad vehicular en términos de una sola denominación, al igual que las demandas de los clientes y la flota de los vehículos es de tipo homogénea. En el trabajo de maestría desarrollado por De La Cruz Hernández (2003), dicho autor adapta estas instancias al agregarle una segunda denominación en términos de capacidad vehicular y demanda de los clientes para hacer referencia a otra medida de carga (peso y volumen, por ejemplo) y también modifica las capacidades de carga de los vehículos al reducir sus valores con el fin de obtener una flota heterogénea e inducir la máxima utilización vehicular. Por otro lado, las distancias (costos)  $c_{ij}^k$  o tiempos de viaje  $t_{ij}^k$ , donde  $t_{ij}^k = c_{ij}^k = c_{ij} \forall k \in K$ , están expresadas en términos de distancia euclidiana y el tiempo de servicio resulta ser  $s_i = s, \forall i \in N \cup \{0\}$ . En el Anexo 2. **Análisis de varianza.**

Al desarrollar un análisis de varianza a la Distancia Total Recorrida para la instancia R201 se observa que el modelo resulta ser no significativo y que ninguno de los parámetros de entrada definidos tiene influencia en el valor de Distancia Total Recorrida. Además el valor R-Cuadrado resulta ser muy bajo. Por otro lado en cuanto al análisis de varianza para el Tiempo de Ejecución, se obtiene un modelo significativo, donde las variables Radio de Giro, Probabilidad de Mutación Reversa y la combinación de las Probabilidades de Mutación Reversa y Espontánea son las que más influyen en el resultado de la variable Tiempo de Ejecución. Por otro lado, aun cuando el modelo es significativo, el valor de R-Cuadrado resulta ser bajo. Lo anterior se puede observar en las siguientes tablas.

Análisis de varianza para Distancia Total Recorrida.

Source	Sum of	DF	Mean	F Value	p-value
--------	--------	----	------	---------	---------

	Squares		Square		Prob > F
Model	8696.117	19	457.690	1.400	0.168
A - Radio de Giro	1232.433	2	616.217	1.885	0.162
B - Largo de Carrera	1137.715	2	568.858	1.740	0.186
C-Prob. Mut. Esp	695.086	1	695.086	2.126	0.151
D-Prob. Mut. Rev	727.266	1	727.266	2.224	0.142
AB	1903.342	4	475.835	1.455	0.229
AC	49.829	2	24.914	0.076	0.927
AD	955.269	2	477.635	1.461	0.241
BC	1506.392	2	753.196	2.304	0.110
BD	455.912	2	227.956	0.697	0.503
CD	32.873	1	32.873	0.101	0.753
Residual	17002.312	52	326.968		
Cor Total	25698.429	71			
R-Squared	0.338				

#### Análisis de varianza para Tiempo de Ejecución.

Source	Sum of Squares	DF	Mean Square	F Value	p-value Prob > F
Model	27195.361	19	1431.335	3.037	0.001
A - Radio de Giro	6819.111	2	3409.556	7.235	0.002
B - Largo de Carrera	736.361	2	368.181	0.781	0.463
C-Prob. Mut. Esp	1.389	1	1.389	0.003	0.957
D-Prob. Mut. Rev	9384.500	1	9384.500	19.913	< 0.0001
AB	4757.889	4	1189.472	2.524	0.052
AC	797.444	2	398.722	0.846	0.435
AD	976.000	2	488.000	1.036	0.362
BC	130.194	2	65.097	0.138	0.871
BD	175.583	2	87.792	0.186	0.831
CD	3416.889	1	3416.889	7.250	0.010
Residual	24505.750	52	471.264		
Cor Total	51701.111	71			
R-Squared	0.526				



**Anexo 3** y **Anexo 4** se encuentran las instancias R201 y R209 respectivamente.

En cuanto al algoritmo bacterial, se llevaron a cabo una serie corridas de manera aleatoria con cierto grupo de parámetros, a fin de observar los resultados obtenidos en los objetivos a minimizar en el problema, al igual que el tiempo de ejecución del algoritmo, el cual fue desarrollado en lenguaje Visual Basic. Se utilizó la herramienta Design Expert 8.0 Trial para la elaboración del orden de las corridas y los parámetros para cada una de estas, y dichas corridas se realizaron en un equipo portátil con procesador Intel Core I5 de 2.67 GHz y memoria RAM de 4 GB.

## **6.1. DEFINICIÓN DE PARAMETROS**

De acuerdo con Gómez Vizcaíno (2009), se conoce que si la energía inicial no supera el total de los desgastes normales que sufre la bacteria a través del ciclo, es decir, si la sumatoria de los desgastes por giro, carrera y fisión supera a la energía inicial, la cantidad de soluciones (bacterias) podrían llegar al punto de desaparecer y la energía por fotosíntesis debe ser por lo menos la suma de todos los desgastes; por último la población de soluciones puede crecer mucho si se asigna una cantidad de energía inicial muy alta. Por otro lado, la cantidad de soluciones iniciales es un factor que influye en gran medida en el tiempo de ejecución del algoritmo, por lo que este valor se fijará en cinco, ya que con esta cantidad se han encontrado soluciones de buena calidad en tiempos no muy altos. En cuanto a la cantidad de ciclos bacterianos, también influye en el tiempo de ejecución del algoritmo, pero se realizaron algunas corridas y se observó que después de 2000 ciclos no existía una mejora significativa en la calidad de las soluciones. Un fenómeno similar se observó con la relación entre la cantidad de ciclos de reducción de rutas y el tiempo de ejecución del algoritmo la cual resultó ser directamente proporcional, pero al realizar algunas corridas se encontró que con tres ciclos se lograba encontrar la cantidad mínima de vehículos reportados para las instancias R201 y R209.

Teniendo en cuenta lo anterior se han seleccionado cuatro factores (parámetros) principales que se presentan a continuación.

### **Parámetros y valores.**

Los factores escogidos con sus respectivos niveles a evaluar son los siguientes:

- Radio de giro – *radio\_de\_giro* (A): 4, 6 y 8
- Largo de carrera – *largo\_carrera* (B): 400, 450 y 500
- Probabilidad de mutación espontánea (C): 0.2 y 0.8
- Probabilidad de mutación reversa (D): 0.2 y 0.8

Las respuestas a analizar son:

- Distancia total recorrida - *TTDS<sub>s</sub>*
- Tiempo de ejecución del algoritmo.

## **6.2. ANÁLISIS DE RESULTADOS.**

Para cada combinación de parámetros se realizaron dos replicas obteniéndose un total de 72 corridas. Los resultados para la instancia R201 y R209 se observan en la Tabla 1. Nuevamente se aclara que la cantidad de vehículos obtenidos en cada una de estas corridas es igual a cuatro y tres vehículos para las instancias R201 y R209 respectivamente, que es la menor cantidad de vehículos reportada hasta el momento para cada una de estas instancias.

En la Tabla 2 está consignado un resumen de los resultados obtenidos para las instancias R201 y R209. En este resumen se especifican los límites del intervalo de confianza para la Cantidad de Vehículos y Distancia Total Recorrida en cada una de las instancias con un nivel de confianza del 95%, al igual que la mejor y peor solución encontrada para cada instancia.

Aunque no es parte del alcance de esta investigación, en el Anexo 2 se presenta un breve análisis de varianza para los resultados obtenidos de Distancia Total Recorrida y Tiempo de Ejecución para la instancia R201.

Corrida	PARAMETROS				INSTANCIA R201			INSTANCIA R209		
	Radio de giro: A	Largo de Carrera: B	Probabilidad de mutación espontanea: C	Probabilidad de mutación reversa: D	Cantidad de Vehículos	Distancia Total Recorrida	Tiempo de ejecución (segundos)	Cantidad de Vehículos	Distancia Total Recorrida	Tiempo de ejecución (segundos)
1	4	500	0.2	0.2	4	1372.19	43	3	1004.85	137
2	6	500	0.8	0.2	4	1390.35	61	3	1020.59	158
3	8	400	0.8	0.2	4	1348.88	74	3	1016.18	159
4	8	400	0.8	0.2	4	1367.45	62	3	1056.41	167
5	6	500	0.2	0.2	4	1364.23	46	3	1075.47	108
6	4	450	0.8	0.2	4	1387.79	50	3	1050.04	85
7	8	400	0.2	0.2	4	1354.16	77	3	991.02	146
8	6	400	0.8	0.8	4	1368.74	47	3	1000.53	180
9	4	500	0.8	0.2	4	1366.79	70	3	1004.23	150
10	8	400	0.8	0.8	4	1343.28	155	3	1022.55	158
11	4	450	0.8	0.8	4	1367.89	50	3	1095.82	79
12	8	450	0.8	0.2	4	1369.50	62	3	1049.86	128
13	8	500	0.2	0.8	4	1353.64	93	3	1043.90	216
14	8	500	0.8	0.2	4	1347.81	69	3	994.42	253
15	8	500	0.8	0.8	4	1371.26	76	3	1036.15	138
16	4	500	0.8	0.8	4	1377.51	56	3	1011.88	98
17	4	450	0.8	0.8	4	1350.02	78	3	1076.20	92
18	6	450	0.8	0.2	4	1319.72	130	3	1010.65	106
19	6	500	0.8	0.8	4	1352.76	86	3	1045.89	131
20	8	500	0.8	0.8	4	1366.57	100	3	1001.72	263
21	4	400	0.8	0.8	4	1296.59	47	3	1068.15	89
22	8	400	0.8	0.8	4	1343.02	123	3	1038.98	173

23	4	400	0.8	0.8	4	1316.94	67	3	1068.28	75
24	6	400	0.8	0.2	4	1349.72	60	3	1029.65	168
25	4	500	0.8	0.8	4	1339.39	71	3	1025.29	153
26	8	450	0.8	0.8	4	1369.50	73	3	1006.68	251
27	6	450	0.2	0.2	4	1353.13	59	3	1043.88	165
28	8	450	0.2	0.2	4	1345.63	75	3	982.16	201
29	6	400	0.8	0.2	4	1332.36	79	3	1073.77	96
30	4	450	0.2	0.2	4	1370.72	51	3	969.98	125
31	6	450	0.8	0.8	4	1355.50	77	3	1069.52	199
32	8	500	0.2	0.8	4	1378.51	75	3	977.01	256
33	8	400	0.2	0.8	4	1369.48	113	3	1027.87	222
34	4	400	0.2	0.8	4	1356.02	76	3	1038.69	84
35	6	450	0.2	0.2	4	1380.65	48	3	1024.70	89
36	4	400	0.2	0.2	4	1371.98	55	3	1060.71	99
37	8	450	0.2	0.8	4	1358.93	107	3	1047.29	170
38	6	450	0.2	0.8	4	1329.41	134	3	1056.68	111
39	6	500	0.2	0.2	4	1386.03	52	3	1010.85	167
40	6	400	0.2	0.8	4	1370.24	53	3	1017.30	103
41	8	400	0.2	0.2	4	1349.00	62	3	996.99	203
42	8	450	0.8	0.8	4	1365.44	147	3	1009.11	174
43	6	450	0.2	0.8	4	1354.28	125	3	1050.22	139
44	4	500	0.2	0.8	4	1335.74	104	3	1082.79	86
45	6	400	0.2	0.8	4	1384.23	69	3	1032.55	202
46	8	400	0.2	0.8	4	1385.89	87	3	1018.24	183
47	6	400	0.2	0.2	4	1339.56	53	3	1083.47	120
48	8	500	0.8	0.2	4	1369.70	113	3	1047.64	156
49	4	400	0.8	0.2	4	1319.79	67	3	1080.12	78
50	4	500	0.8	0.2	4	1350.35	79	3	1021.69	150

51	8	450	0.8	0.2	4	1354.99	68	3	1032.94	159
52	4	400	0.8	0.2	4	1364.15	41	3	1040.46	85
53	6	500	0.2	0.8	4	1335.69	140	3	1059.57	147
54	8	500	0.2	0.2	4	1375.31	57	3	1004.29	161
55	4	400	0.2	0.2	4	1367.43	44	3	1026.71	106
56	6	450	0.8	0.8	4	1358.52	61	3	1004.07	209
57	6	400	0.8	0.8	4	1348.31	68	3	1026.44	169
58	4	450	0.2	0.2	4	1346.05	55	3	1092.05	119
59	6	500	0.8	0.8	4	1326.43	81	3	1047.22	136
60	4	500	0.2	0.8	4	1349.34	103	3	1008.39	131
61	8	450	0.2	0.2	4	1369.15	89	3	1004.89	138
62	4	400	0.2	0.8	4	1307.73	102	3	1091.47	73
63	4	450	0.2	0.8	4	1357.86	51	3	1034.43	106
64	6	400	0.2	0.2	4	1370.69	52	3	1004.97	156
65	6	500	0.8	0.2	4	1352.74	58	3	1062.36	176
66	8	500	0.2	0.2	4	1374.99	63	3	1064.03	205
67	8	450	0.2	0.8	4	1350.19	108	3	1017.30	194
68	4	450	0.2	0.8	4	1372.12	58	3	984.26	133
69	6	500	0.2	0.8	4	1355.90	108	3	1047.09	172
70	6	450	0.8	0.2	4	1373.88	85	3	1028.37	99
71	4	450	0.8	0.2	4	1341.79	72	3	1045.79	150
72	4	500	0.2	0.2	4	1353.04	66	3	1069.69	132

Tabla 1. Resultados de las corridas para las instancias a evaluar.

	Instancia R201			Instancia R209		
	Vehículos	Distancia	Error %	Vehículos	Distancia	Error %
Promedio	4	1356.59	8.32%	3	1034.24	13.76%
Desviación estándar	0	19.02	-	0	29.96	-
Peor Solución	4	1390.35	11.02%	3	1095.82	20.53%
Mejor Solución	4	1296.59	3.53%	3	969.98	6.69%
Límite Inferior IC (95%)	4	1352.20	7.97%	3	1027.32	13.00%
Límite Superior IC (95%)	4	1360.99	8.67%	3	1041.16	14.52%
Mejor solución otras heurísticas	4	1252.37	0.00%	3	909.16	0.00%

Tabla 2. Resumen de resultados obtenidos para las instancias R201 y R209

Para la instancia R209 se observa que en promedio se obtienen soluciones con un 13.76% de margen de error relativo con respecto a la mejor solución reportada para esta instancia. Aun cuando la peor solución encontrada se encuentra muy distante con respecto a la mejor solución encontrada por otras heurísticas, se resalta la mejor solución encontrada, con un error relativo de 6.69%, la cual resulta ser buena en términos de calidad de solución. Para la instancia R201, las soluciones en promedio tienen un error relativo del 8.32% y para esta instancia la peor solución encontrada no supera el 12% y la mejor solución tiene un error relativo del 3.53%, concluyéndose que para esta instancia la calidad de los resultados obtenidos es alta. En síntesis, para las instancias en que fue probada esta metodología y sin haberse realizado una configuración adecuada de parámetros (optimización de parámetros), se logró obtener en promedio soluciones con un error relativo inferior al 15% con respecto a la mejor solución reportada para las instancias seleccionadas.

La calidad de los mejores resultados obtenidos para cada instancia al compararse con los obtenidos por la alternativa de dos fases desarrollada por De La Cruz Hernández (2003), la cual da solución al problema planteado en esta investigación, se observa que se lograron obtener mejores resultados que esta propuesta, especialmente en la instancia R201 donde la diferencia entre el mejor resultado reportado por la Metaheurística Bacterial y la Alternativa de dos fases es bastante notoria. Lo anterior se puede observar en la Tabla 3.

	Instancia R201			Instancia R209		
	Vehículos	Distancia	Error %	Vehículos	Distancia	Error %
Mejor solución otras heurísticas	4	1252.37	0.00%	3	909.16	0.00%
Metaheurística Bacterial	4	1296.59	3.53%	3	969.98	6.69%
Alternativa de 2 fases (De La Cruz Hernández 2003)	4	1385.28	10.61%	3	975.47	7.29%

Tabla 3. Comparación de la metodología propuesta con otras heurísticas.

En el Anexo 5 y Anexo 6 se pueden ver las mejores soluciones encontradas para las instancias R201 y R209 respectivamente.

### 6.2.1. Tiempo computacional.

	Tiempo (segundos)	
	Instancia R201	Instancia R209
Promedio	76.61	147.15
Desviación estándar	26.98	46.89
Peor Tiempo Registrado	155.00	263.00
Mejor Tiempo Registrado	41.00	73.00
Límite Inferior IC (95%)	70.38	136.32
Límite Superior IC (95%)	82.84	157.98

Tabla 4. Tiempos de ejecución del algoritmo en las instancias seleccionadas.

Los tiempos de ejecución del algoritmo para la solución de cada instancia están consignados en la Tabla 4. Los tiempos resultantes son altos pero estos tiempos se explican por las características de la metodología propuesta. Inicialmente influye la cantidad de soluciones iniciales con las que se ejecutaron las corridas del programa, pero debe tenerse en cuenta que en la eliminación de rutas se generan soluciones no factibles y que deben emplearse esfuerzos por retornar la factibilidad de cada solución, lo cual consume una cantidad significativa de tiempo. Por otra parte, la metaheurística bacterial genera un crecimiento exponencial de las soluciones, y aun cuando se controlaron los parámetros relacionados con la energía para evitar un crecimiento desmesurado de las soluciones,



durante la ejecución de los ciclos bacterianos se pueden encontrar entre cinco y diez soluciones a procesar por cada ciclo y esto repercute en los resultados obtenidos para cada instancia. En sí, los resultados de tiempo no son comparables con otros autores puesto que sus algoritmos fueron desarrollados bajo otras condiciones.

## 7. CONCLUSIONES Y FUTURAS INVESTIGACIONES

Al observar todo lo desarrollado en esta investigación, se concluye que con esta se logró desarrollar un algoritmo basado en el fototaxismo de las bacterias el cual fue adaptado a los objetivos del problema HFMPVRPTW, además de complementarlo con otros conceptos que se han desarrollado para la solución de problemas de enrutamiento de vehículos. La calidad de los resultados arrojados por la metodología fue posible compararse con los mejores resultados obtenidos para las instancias en las que se probó la metodología, al igual que con otro trabajo que directamente propone una metodología para resolver el HFMPVRPTW. Lo anterior ha sido plasmado en este documento y cualquier lector puede ver claramente las características de esta metodología. En síntesis, se ha dado cumplimiento con los objetivos específicos planteados en esta investigación lo que conlleva a concluir que se ha cumplido con el objetivo general al lograr diseñar y desarrollar la metodología de optimización propuesta para el problema en cuestión.

Aunque el objetivo general de la investigación fue alcanzado, se hace necesario hacer algunos comentarios con respecto a la metodología. El primero de estos hace referencia a la construcción inicial de la ruta la cual no fue desarrollada con los métodos clásicos creados para este fin, sino que se utilizó la fórmula  $GLS_i$  para la aplicación de la rutina Giro\_Bacterias, fórmula característica de la metaheurística bacteriana (Gómez Vizcaíno 2009). Para las instancias en que fue probada esta construcción, se obtuvieron soluciones cuya cantidad de vehículos iniciales eran muy cercanas a la menor cantidad de vehículos reportada para cada instancia, y que conllevaba a tener que realizar pocos ciclos de eliminación de rutas para alcanzar este objetivo a minimizar. En las corridas realizadas se propuso fijar el valor de ciclos para eliminación de rutas a tres a fin que fuera un valor estándar para ambas instancias, sin embargo en pruebas realizadas puntualmente para cada instancia se encontró que para la instancia R201 se podía llegar al menor número de vehículos con solo aplicar dos ciclos mientras que para la R209 se llegaba con un solo

ciclo. En resumen, la creación inicial de la ruta desarrollada en esta investigación es susceptible de analizarse y realizar mejoras, la cual se propone como un tema de investigación futura. Esto último también aplica a la eliminación de rutas que se empleó en esta metodología, ya que dicha eliminación solo tenía en cuenta la ruta más pequeña de la solución y no evaluaba a las restantes, por lo que esto podría ser una idea para mejorar la técnica.

El algoritmo bacterial se caracteriza por tener un crecimiento exponencial de las soluciones y que tiene una gran influencia en los tiempos de ejecución del algoritmo, al igual que la eliminación de rutas (generación de soluciones no factibles). Para contrarrestar lo anterior, se siguieron las recomendaciones del autor del algoritmo bacterial y se fijaron unos valores de energía de tal manera que no existiese un crecimiento desbordado de las soluciones. Por otro lado, para la generación de vecindades solo se utilizó la técnica de *inserción* tanto en los movimientos realizados por las bacterias como en la *mutación*. Estos dos aspectos son susceptibles de análisis para realizar mejoras a la metodología y que sirven de base a futuras investigaciones.

El tiempo computacional empleado por este algoritmo resulta ser alto, fenómeno explicado por el crecimiento exponencial de las bacterias y la generación de soluciones no factibles en la eliminación de rutas. No obstante, la calidad de las soluciones obtenidas en las instancias utilizadas es muy buena, especialmente en la instancia R201 al encontrarse una solución con un error relativo inferior al 4%. Por otro lado, los tiempos de convergencia hacia el alcance de las soluciones entre las instancias resultan ser muy distintos, quizás esto se explique por las características propias de la instancia, haciendo que en la instancia R201 se emplee un vehículo más que en la instancia R209 y por consiguiente en la instancia R201 se puedan acomodar los clientes de una manera mejor y más rápida. En conclusión a esto, la persona encargada de la toma de decisiones para resolver el problema debe hacer una evaluación entre calidad de solución versus tiempo para así seleccionar la metodología que más le convenga.

En cuanto a las corridas realizadas para la evaluación de la calidad del algoritmo, se emplearon unos valores entre los cuales se lograron observar buenas soluciones para cada instancia. Esta investigación no tenía como objetivo realizar un análisis experimental de la configuración de los parámetros de entrada al algoritmo, tema que se propone como investigación futura para determinar los valores más apropiados.

Por último, además de las anteriores propuestas de investigación, se propone la aplicación de esta metodología en la solución de un problema real, al igual que en instancias de tipo estocástico y dinámico.

## REFERENCIAS BIBLIOGRAFICAS.

Alba, Enrique. *Parallel Metaheuristics: New Class of Algorithms*. New Jersey: Wiley Interscience, 2005.

Alpaydin, Ethem. *Introduction to Learning Machine*. Segunda edición. Londres: The MIT Press, 2010.

Alvarenga, G. B., G. R. Mateus, y G. De Tomi. «A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows.» *Computers & Operations Research*, n° 34 (2007): 1561–1584.

Aronson, L.D. «Algorithms for vehicle routing - A survey.» Reporte, Faculty of Technical Mathematics and Informatics, Delft University of Technology, 1995.

Ayers, James. *Handbook of Supply Chain Management*. Virginia: The St. Lucie Press, 2001.

Badeau, Philippe, François Guertin, Michel Gendreau, Jean-Yves Potvin, y Éric Taillard. «A parallel tabu search heuristic for the vehicle routing problem with time windows.» *Transportation Research Part C: Emerging Technologies* V, n° 2 (1997): 109-122.

Baker, E., y J. Schaffer. «Solution improvement heuristics for the vehicle routing and scheduling problem with time windows constraints.» *American Journal of Mathematics and Management Sciences*, 1989: 261-300.

Ballou, Ronald. *Logística: Administración de la Cadena de Suministro*. Quinta edición. Mexico: Pearson - Prentice Hall, 2004.

Bent, Russell, y Pascal Van Hentenryck. «A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows.» *Computers & Operations Research*, n° 33 (2006): 875–893.

Berger, Jean, y Mohamed Barkaoui. «A parallel hybrid genetic algorithm for the vehicle routing problem with time windows.» *Computers & Operations Research*, n° 31 (2004): 2037–2053.

Birbil, S., y Sh. Fang. «An Electromagnetism-like Mechanism for Global Optimization.» *Journal of Global Optimization* XV (2003): 263-282.

Bramel, Julien, y David Simchi-Levi. *The logic of logistics*. New York: Springer - Verlag, 1997.

Bramel, Julien., y David. Simchi-Levi. «A location based heuristic for the vehicle routing problem.» *Operation Research*, n° 43 (1995): 649-660.

Bräysy, Olli, Wout Dullaert, y Michel Gendreau. «Evolutionary algorithms for the vehicle routing problem with time windows.» *Journal of Heuristics*, n° 159 (2004a): 586-605.

Bräysy, Olli, y Wout Dullaert. «A fast evolutionary metaheuristic for the vehicle routing problem with time windows.» *International Journal on Artificial*, n° 12 (2003): 153-172.

Bräysy, Ollit, y Michel Gendreau. «Genetic algorithms for the vehicle routing problems with time windows.» Reporte interno, Departamento de Optimización, SINTEF Applied Mathematics, Oslo, 2001.

Bräysy, Ollit, y Michel Gendreau. «Vehicle routing problem with time windows, part II: metaheuristics.» *Transportation Science*, n° 39 (2005b): 119-139.

Chin, Ang Juay, Ho Wee Kit, y Andrew Lim. «A new GA approach for vehicle routing problem.» *Proceedings of IEEE International Conference on Tools with Artificial Intelligence*. 1999. 307-310.

Choia, Eunjeong, y Dong-Wan Tchab. «A column generation approach to the heterogeneous fleet vehicle routing problem.» *Computers & Operations Research*, n° 34 (2007): 2080-2095.

Clarke, G., y J. W. Wright. «Scheduling of vehicles from a central depot to a number of delivery points.» *Operation Research*, n° 12 (1964): 568-581.

Cordeau, Jean-François, Gilbert Laporte, y A. Mercier. «unified tabu search heuristic for the vehicle routing problems with time windows.» *The Journal of Operation Research Society LII*, n° 8 (2001): 928-936.

Dasgupta, D. *Artificial Immune Systems and Their Applications*. Berlin: Springer- Verlag, 1999.

De La Cruz Hernández, Jair. «Una alternativa heurística de dos fases para el problema de enrutamiento de vehículos con ventanas de tiempo, múltiples productos y flota heterogénea.» Tesis de Maestría, Departamento de Ingeniería Industrial, Universidad del Norte, Barranquilla, 2003.

De La Cruz, Jair, Carlos D. Paternina-Arboleda, Victor Cantillo, y Jairo R. Montoya-Torres. «A two-pheromone trail ant colony system-tabu search approach for the

heterogeneous vehicle routing with time windows and multiple products.» *Journal of Heuristics*, 2011.

Dondo, Rodolfo, y Jaime Cerdá. «A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows.» *European Journal of Operational Research*, nº 176 (2007): 1478–1507.

Donoso, Yesid, y Ramon Fabregat. *Multi-objective optimization in computer networks using metaheuristics*. Florida: Auerbach Publications, 2007.

Dorigo, Marco, y T. Stutzle. *Ant Colony Optimization*. Prentice Hall, 2004.

Eberhart, C., y J. Kennedy. «A new optimizer using particle swarm theory.» *Proceedings of Sixth International Symposium of Micro Machine and Human Science*. Nagoya, 1995. 39-43.

Ferlan, J. A., y P. Michelon. «The vehicle scheduling problem with multiple vehicle types.» *Journal of Operation Research Society*, nº 39 (1988): 577-583.

Fiechter, C. N. «A parallel tabu search algorithm for large traveling salesman problems.» *Discrete Applied Mathematics*, LI, nº 3 (1994): 243-267.

Fisher, M, y R. Jaikumar. «A generalized assignment heuristic for the vehicle routing problem.» *Networks*, nº 11 (1981): 109-124.

Gambardella, Luca M., Éric Taillard, y Giovanni. Agazzi. «MACS-VRPTW: Vehicle routing problem with time window.» En *New Ideas in Optimization*, de D. Corne, M. Dorigo y F. Glover, 63-76. Londres: McGraw-Hill, 1999.

Garcia, Bruno-Laurent, Jean-Yven Potvin, y Jean-Marc Rousseau. «A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints.» *Computers & Operation Research* XXI, nº 9 (1994): 1025-1033.

Garey, Michael, y David Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

Gómez Vizcaíno, Libardo Segundo. «Algoritmo de optimización multiobjetivo basado en la fototaxis de las bacterias (GLSBA: Gómez Libardo's Bacterial Algorithm).» Tesis de Maestría, Departamento de Ingeniería Industrial, Universidad del Norte, Barranquilla, 2009.

Guillet, B., y L. Miller. «A heuristic algorithm for the vehicle dispatch problem.» *Operation Research* XXII (1974): 340-349.

Gutin, Gregory, y Abraham P. Punnen. *The Travel Salesman Problem and Its Variations*. Vol. XII. New York: Kluwer Academic Publisher, 2004.

Haralambos, Sarimveis, Panagiotis Patrinos, Chris Tarantilis, y Chris Kiranoudis. «Dynamic Modeling and Control of Supply Chain Systems: A Review.» *Computers & Operations Research* XXXV, n° 11 (2008): 3530-3561.

Helsgaun, Keld. «An effective implementation of the Lin±Kernighan traveling salesman heuristic.» *European Journal of Operational Research*, n° 126 (2000): 106-130.

Ho, S. C., y D. Haugland. «A tabu search heuristic for the vehicle routing problem with time windows and split deliveries.» *Computers & Operations Research*, n° 31 (2004): 1947–1964.

Homburger, Jörg, y Hermann Gehring. «A two-phase hybrid metaheuristic for the vehicle routing problem with time windows.» *European Journal of Operational Research*, n° 162 (2005): 220-238.

Kirkpatrick, S., C. Gelatt, y M. P. Vecchi. «Optimization by simulated annealing.» *Science* CCXX, n° 4598 (1983): 671-680.

Liu, Fuh-Hwa, y Sheng-Yuan Shen. «A method for vehicle routing problem with multiple vehicle types and time windows.» *Proceedings of National Science Council* XXIII, n° 4 (1998): 526-536.

Nagata, Yuichi, y Olli Bräysy. «A powerful route minimization heuristic for the vehicle routing problem with time windows.» *Operations Research Letters*, n° 37 (2009): 333-338.

Potvin, Jean-Yves, y S. Bengio. «The vehicle routing problem with time windows – part II: Genetic search.» *INFORMS Journal on Computing* VIII (1996): 165-172.

Potvin, Jean-Yves, y Jean-Marc Rousseau. «A parallel route building for the vehicle routing and scheduling problem with time windows.» *European Journal of Operational Research* LXVI, n° 3 (1993): 331-340.

Potvin, Jean-Yves, y Jean-Marc Rousseau. «An exchange heuristic for routing problems with time windows.» *Journal of Operational Research Society* XLVI (1995): 1433-1446.

Russell, Robert. «Hybrid heuristics for the vehicle routing problem with time windows.» *Transportation Science* XXIX, n° 2 (1995): 156-166.

Russell, Robert, y Wen-Chyuan Chiang. «A metaheuristic for the vehicle routing problem with soft time windows.» *The Journal of Operation Research Society* LV, n° 12 (2004): 1298-1310.



Russell, Robert, y Wen-Chyuan Chiang. «Simulated annealing metaheuristics for the vehicle routing problem with time windows.» *Annals of Operations Research* LXIII (1996): 3-27.

Schulze, J., y T Fahle. «A parallel algorithm for the vehicle routing problem with time window constraints.» *Annals of Operation Research*, n° 86 (1999): 575-607.

Shah-Hosseini, H. «Problem solving by intelligent water drops.» *2007 IEEE Congress on Evolutionary Computation (CEC 2007)*. 2007. 3226-3231.

Solomon, Marius. «Algorithms for the vehicle routing and scheduling problem with time windows.» *Operations Research* XXXV, n° 2 (1987): 254-265.

—. *VRPTW Benchmark Problems*. 2005. <http://web.cba.neu.edu/~msolomon/problems.htm> (último acceso: Mayo de 2010).

Sutton, Richard S., y Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge: The MIT Press, 1998.

Tan, K. C., L. H. Lee, Q. L. Zhu, y K. Ou. «Heuristic methods for vehicle routing problem with time windows.» *Artificial Intelligence in Engineering* XV (2001): 281-295.

Tan, Xuan, Xuyao Luo, W.N. Chen, y Jun Zhang. «Ant Colony System for Optimizing Vehicle Routing Problem with Time Windows.» *International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference of Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)*. 2005.

Tavakkoli-Moghaddam, R., N. Safaei, y Y. Gholipour. «A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length.» *Applied Mathematics and Computation*, n° 176 (2006): 445-454.

Teodorovic, D., P. Lucic, G. Markovic, y M. Dell'Orco. «Bee colony optimization: principles and applications.» *8th Seminar on Neural Networks Applications in Electrical Engineering, NEUREL-2006*. 2006.

Thangiah, Sam. *Vehicle Routing with Time Windows using Genetic Algorithms*. Vol. II, de *Applications Handbook of Genetic Algorithms*., editado por L. Chambers, 253-277. Boca Raton, Florida: CRC Press, 1995.

Thangiah, Sam, Ibrahim Osman, y Tong Sun. «Hybrid genetic algorithm, simulated annealing and tabu search methods for vehicle routing problems with time windows.» Reporte técnico, Computer Science Department, Slippery Rock University, 1994.

Thangiah, Sam, Ibrahim Osman, y Tong Sun. «Metaheuristics for the vehicle routing problem with time windows.» Reporte técnico SUR-CpSc-TR-95-32, Artificial Intelligence

and Robotics Laboratory, Computer Science Department, Slippery Rock University, Pennsylvania, 1995.

Toth, Paolo, y Daniele Vigo. *The Vehicle Routing Problem*. Philadelphia: SIAM, 2002.

Van Landeghem, H. «A bi-criteria heuristic for the vehicle routing problem with time windows.» *European Journal of Operational Research* XXXVI, nº 2 (1988): 217-226.

## ANEXOS

Anexo 1. Formula Giro Bacteria.

El autor de la metaheurística bacterial propone la siguiente expresión:

$$GLS_i = \frac{\prod_{r=1}^{\frac{nobj}{2}} e^{\left(\frac{-f_{2*r-1}(B_i)}{\max\{f_{2*r-1}(C)\}}\right)}}{\prod_{r=1}^{\frac{nobj}{2}} e^{\left(\frac{f_{2*r}(B_i)}{\max\{f_{2*r}(C)\}}\right)}}, B_i \in C \quad (4)$$

La expresión (4) es equivalente a la siguiente:

$$GLS_i = e^{-\sum_{r=1}^{\frac{nobj}{2}} \left(\frac{f_{2*r-1}(B_i)}{\max\{f_{2*r-1}(C)\}}\right)} * e^{-\sum_{r=1}^{\frac{nobj}{2}} \left(\frac{f_{2*r}(B_i)}{\max\{f_{2*r}(C)\}}\right)}, B_i \in C$$

$$GLS_i = e^{-\sum_{r=1}^{\frac{nobj}{2}} \left(\frac{f_{2*r-1}(B_i)}{\max\{f_{2*r-1}(C)\}} + \frac{f_{2*r}(B_i)}{\max\{f_{2*r}(C)\}}\right)}, B_i \in C \quad (5)$$

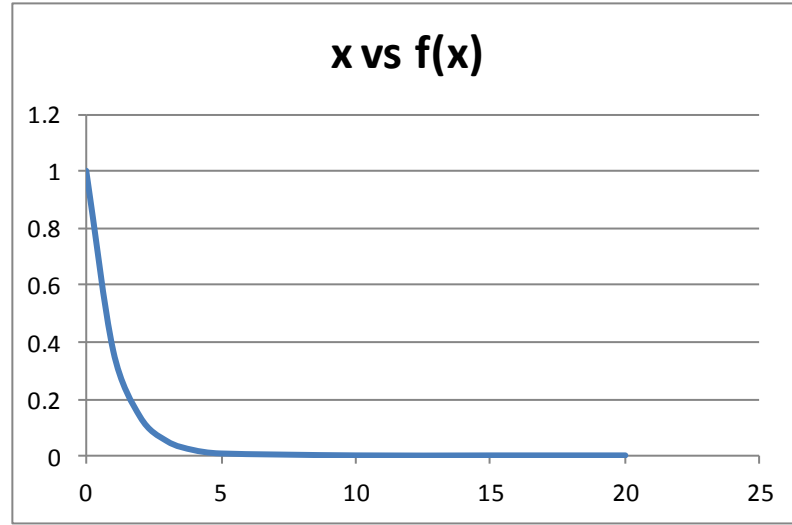
La expresión (5) resulta extraña cuando  $nobj = 1$ , al igual que si la variable  $nobj$  es un número impar ya que la expresión  $\frac{f_{nobj-1}}{\max\{f_{nobj-1}\}}$  se suma dos veces. Por lo tanto, para dar igual ponderación a cada objetivo se propone lo siguiente:

$$GLS_i = e^{-\sum_{r=1}^{nobj} \frac{f_r(B_i)}{\max\{f_r(C)\}}}, B_i \in C \quad (6)$$

Se aclara que esta última expresión no es la definitiva, ya que no es motivo de esta investigación presentar un análisis y demostración formal de dicha expresión. Por otro lado, las ecuaciones (5) y (6) son semejantes a una distribución exponencial, siendo esta una distribución de probabilidad continua con parámetro  $\lambda > 0$  definida de la siguiente manera:

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \forall x \geq 0 \\ 0 & \text{otro caso} \end{cases} \quad (7)$$

En la grafica  $x \text{ vs } f(x)$  con  $\lambda = 1$ , se observa que a medida que el valor de  $x$  es más pequeño el valor de  $f(x) \approx 1$ .



Un ejemplo para esto es el siguiente: suponga que se desea minimizar o maximizar las expresiones  $f_1(x, y) = (x - 2)^2 + (y - 2)^2$  y  $f_2(x, y) = x^2 + y^2$ . Para esto se utilizara la expresión  $GLS_i$  de la siguiente forma:

$$GLS_i = \frac{e^{\left(\frac{-f_1(x,y)}{\max\{f_1(x,y)\}}\right)}}{e^{\left(\frac{f_2(x,y)}{\max\{f_2(x,y)\}}\right)}} = e^{\left(\frac{-f_1(x,y)}{\max\{f_1(x,y)\}}\right)} * e^{\left(\frac{-f_2(x,y)}{\max\{f_2(x,y)\}}\right)}$$

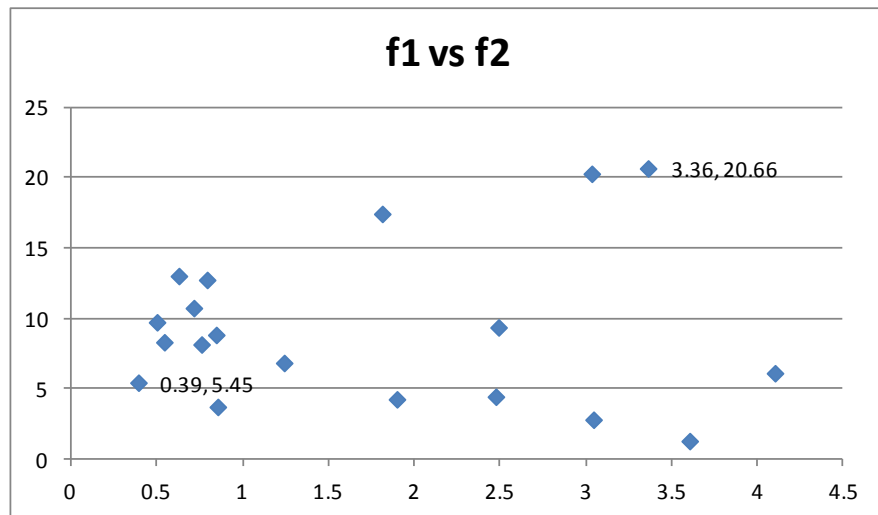
$$GLS_i = e^{\left(\frac{-f_1(x,y)}{\max\{f_1(x,y)\}}\right) + \left(\frac{-f_2(x,y)}{\max\{f_2(x,y)\}}\right)} = e^{-\sum_{r=1}^2 \left(\frac{f_r(x,y)}{\max\{f_r(x,y)\}}\right)} \quad (8)$$

En la tabla de resultados se observan una serie de pares  $(x, y)$  y sus resultados en los objetivos a analizar, al igual que el valor de  $GLS_i$  para cada par. Si se desea minimizar  $f_1$  y  $f_2$  se debe seleccionar aquellos pares que den como resultado un valor de  $GLS_i$  cercano a 1, mientras que si se desea maximizar  $f_1$  y  $f_2$  deben ser valores de  $GLS_i$  cercanos a 0. Lo anterior también puede observarse en la gráfica de  $f_1$  vs  $f_2$ . Se deja abierta la investigación relacionada con la minimización de unos objetivos y maximización de otros.

Tabla de resultados

$x$	$y$	$f_1$	$f_2$	$GLS_i$
3.7373	2.5862	3.3618	20.6557	0.1621
2.7287	3.5820	3.0337	20.2764	0.1788
0.0311	2.4742	4.1015	6.1226	0.2735
2.9529	2.9510	1.8124	17.4279	0.2765
0.7533	2.9674	2.4902	9.3727	0.3461
1.0935	0.3319	3.6043	1.3060	0.3899
1.6547	0.2898	3.0439	2.8220	0.4153
2.0665	0.4283	2.4747	4.4538	0.4409
2.8836	2.1041	0.7917	12.7425	0.4449
2.4389	2.6591	0.6271	13.0191	0.4570
2.7949	1.7130	0.7143	10.7461	0.4994
0.6226	1.9702	1.8980	4.2693	0.5120
0.9724	2.4298	1.2407	6.8496	0.5304
2.6492	1.3493	0.8448	8.8386	0.5305
2.6299	1.6794	0.4996	9.7370	0.5526
2.5368	1.3137	0.7591	8.1611	0.5598
1.4523	2.4923	0.5422	8.3207	0.5857
1.4889	1.2302	0.8538	3.7302	0.6779
1.8781	1.3866	0.3912	5.4498	0.6982

Grafica  $f_1$  vs  $f_2$



## Anexo 2. Análisis de varianza.

Al desarrollar un análisis de varianza a la Distancia Total Recorrida para la instancia R201 se observa que el modelo resulta ser no significativo y que ninguno de los parámetros de entrada definidos tiene influencia en el valor de Distancia Total Recorrida. Además el valor R-Cuadrado resulta ser muy bajo. Por otro lado en cuanto al análisis de varianza para el Tiempo de Ejecución, se obtiene un modelo significativo, donde las variables Radio de Giro, Probabilidad de Mutación Reversa y la combinación de las Probabilidades de Mutación Reversa y Espontánea son las que más influyen en el resultado de la variable Tiempo de Ejecución. Por otro lado, aun cuando el modelo es significativo, el valor de R-Cuadrado resulta ser bajo. Lo anterior se puede observar en las siguientes tablas.

### Análisis de varianza para Distancia Total Recorrida.

Source	Sum of Squares	DF	Mean Square	F Value	p-value Prob > F
Model	8696.117	19	457.690	1.400	0.168
A - Radio de Giro	1232.433	2	616.217	1.885	0.162
B - Largo de Carrera	1137.715	2	568.858	1.740	0.186
C-Prob. Mut. Esp	695.086	1	695.086	2.126	0.151
D-Prob. Mut. Rev	727.266	1	727.266	2.224	0.142
AB	1903.342	4	475.835	1.455	0.229
AC	49.829	2	24.914	0.076	0.927
AD	955.269	2	477.635	1.461	0.241
BC	1506.392	2	753.196	2.304	0.110
BD	455.912	2	227.956	0.697	0.503
CD	32.873	1	32.873	0.101	0.753
Residual	17002.312	52	326.968		
Cor Total	25698.429	71			
R-Squared	0.338				

Análisis de varianza para Tiempo de Ejecución.

Source	Sum of Squares	DF	Mean Square	F Value	p-value Prob > F
Model	27195.361	19	1431.335	3.037	0.001
A - Radio de Giro	6819.111	2	3409.556	7.235	0.002
B - Largo de Carrera	736.361	2	368.181	0.781	0.463
C-Prob. Mut. Esp	1.389	1	1.389	0.003	0.957
D-Prob. Mut. Rev	9384.500	1	9384.500	19.913	< 0.0001
AB	4757.889	4	1189.472	2.524	0.052
AC	797.444	2	398.722	0.846	0.435
AD	976.000	2	488.000	1.036	0.362
BC	130.194	2	65.097	0.138	0.871
BD	175.583	2	87.792	0.186	0.831
CD	3416.889	1	3416.889	7.250	0.010
Residual	24505.750	52	471.264		
Cor Total	51701.111	71			
R-Squared	0.526				

### Anexo 3. Instancia R201 para HFMPVRPTW

Tabla de clientes.

$i$	$x_i$	$y_i$	$p_i$	$v_i$	$e_i$	$l_i$	$s_i$
0	35	35	0	0	0	1000	0
1	41	49	10	18	707	848	10
2	35	17	7	8	143	282	10
3	55	45	13	12	527	584	10
4	55	20	19	14	678	801	10
5	15	30	26	12	34	209	10
6	25	30	3	4	415	514	10
7	20	50	5	16	331	410	10
8	10	43	9	18	404	481	10
9	55	60	16	18	400	497	10
10	30	60	16	20	577	632	10
11	20	65	12	21	206	325	10
12	50	35	19	14	228	345	10
13	30	25	23	9	690	827	10
14	15	10	20	18	32	243	10
15	30	5	8	12	175	300	10
16	10	20	19	10	272	373	10
17	5	30	2	12	733	870	10
18	20	40	12	10	377	434	10
19	15	60	17	8	269	378	10
20	45	65	9	17	581	666	10
21	45	20	11	18	214	331	10
22	45	10	18	20	409	494	10
23	55	5	29	15	206	325	10
24	65	35	3	20	704	847	10
25	65	20	6	12	817	956	10
26	45	30	17	15	588	667	10
27	35	40	16	9	104	255	10
28	41	37	16	16	114	255	10
29	64	42	9	12	190	313	10
30	40	60	21	8	259	354	10
31	31	52	27	9	152	275	10
32	35	69	23	17	660	777	10
33	53	52	11	15	45	200	10
34	65	55	14	18	529	614	10
35	63	65	8	11	686	813	10
36	2	60	5	11	41	208	10
37	20	20	8	2	606	693	10
38	5	5	16	8	302	405	10
39	60	12	31	16	33	224	10
40	40	25	9	18	360	437	10
41	42	7	5	16	396	511	10
42	24	12	5	12	25	172	10
43	23	3	7	18	620	705	10
44	11	14	18	18	233	340	10
45	6	38	16	20	29	189	10
46	2	48	1	12	515	628	10
47	8	56	27	13	85	250	10
48	13	52	36	15	773	906	10
49	6	68	30	15	501	540	10
50	47	47	13	24	547	642	10

$i$	$x_i$	$y_i$	$p_i$	$v_i$	$e_i$	$l_i$	$s_i$
51	49	58	10	20	348	453	10
52	27	43	9	15	174	299	10
53	37	31	14	3	414	489	10
54	57	29	18	15	641	734	10
55	63	23	2	6	620	739	10
56	53	12	6	18	585	692	10
57	32	12	7	5	421	530	10
58	36	26	18	16	849	980	10
59	21	24	28	14	17	229	10
60	17	34	3	12	721	862	10
61	12	24	13	18	290	377	10
62	24	58	19	12	163	302	10
63	27	69	10	18	34	191	10
64	15	77	9	18	214	333	10
65	62	77	20	15	49	188	10
66	49	73	25	15	592	693	10
67	67	5	25	12	294	401	10
68	56	39	36	3	637	752	10
69	37	47	6	13	162	293	10
70	37	56	5	12	788	968	10
71	57	68	15	11	268	367	10
72	47	16	25	18	31	208	10
73	44	17	9	12	308	399	10
74	46	13	8	12	681	802	10
75	49	11	18	21	236	345	10
76	49	42	13	15	290	373	10
77	53	43	14	10	817	952	10
78	61	52	3	6	384	499	10
79	57	48	23	15	388	465	10
80	56	37	6	14	839	968	10
81	55	54	26	13	411	456	10
82	15	47	16	14	162	289	10
83	14	37	11	2	96	249	10
84	11	31	7	8	436	511	10
85	16	22	41	8	376	461	10
86	4	18	35	15	388	465	10
87	28	18	26	15	420	447	10
88	26	52	9	14	279	388	10
89	26	35	15	8	755	920	10
90	31	67	3	8	392	487	10
91	15	19	1	5	739	866	10
92	22	22	2	9	18	181	10
93	18	24	22	12	811	969	10
94	26	27	27	14	436	503	10
95	25	24	20	20	92	231	10
96	22	27	11	10	607	690	10
97	25	21	12	18	612	673	10
98	19	21	10	9	183	306	10
99	20	26	9	10	333	432	10
100	18	18	17	10	798	965	10



Tabla de Vehículos

$k$	$P_k$	$V_k$
1	900	900
2	900	900
3	900	900
4	900	900
5	900	900
6	750	750
7	750	750
8	750	750
9	750	750
10	750	750
11	600	600
12	600	600
13	600	600
14	600	600
15	600	600
16	450	450
17	450	450
18	450	450
19	450	450
20	450	450
21	300	300
22	300	300
23	300	300
24	300	300
25	300	300

#### Anexo 4. Instancia R209 para HFMPVRPTW

Tabla de clientes.

$i$	$x_i$	$y_i$	$p_i$	$v_i$	$e_i$	$l_i$	$s_i$
0	35	35	0	0	0	1000	0
1	41	49	10	9	636	919	10
2	35	17	7	6	74	351	10
3	55	45	13	12	498	613	10
4	55	20	19	18	470	965	10
5	15	30	26	25	20	370	10
6	25	30	3	2	266	663	10
7	20	50	5	4	292	449	10
8	10	43	9	8	288	597	10
9	55	60	16	15	254	643	10
10	30	60	16	15	496	713	10
11	20	65	12	11	33	510	10
12	50	35	19	18	170	403	10
13	30	25	23	22	426	978	10
14	15	10	20	19	32	454	10
15	30	5	8	7	30	529	10
16	10	20	19	18	222	423	10
17	5	30	2	1	409	959	10
18	20	40	12	11	349	462	10
19	15	60	17	16	215	432	10
20	45	65	9	8	538	709	10
21	45	20	11	10	156	389	10
22	45	10	18	17	280	623	10
23	55	5	29	28	36	513	10
24	65	35	3	2	633	918	10
25	65	20	6	5	400	956	10
26	45	30	17	16	548	707	10
27	35	40	16	15	5	612	10
28	41	37	16	15	44	325	10
29	64	42	9	8	129	374	10
30	40	60	21	20	115	498	10
31	31	52	27	26	91	336	10
32	35	69	23	22	602	835	10
33	53	52	11	10	24	646	10
34	65	55	14	13	487	656	10
35	63	65	8	7	439	948	10
36	2	60	5	4	41	710	10
37	20	20	8	7	563	736	10
38	5	5	16	15	147	560	10
39	60	12	31	30	33	797	10
40	40	25	9	8	243	554	10
41	42	7	5	4	225	682	10
42	24	12	5	4	25	616	10
43	23	3	7	6	578	747	10
44	11	14	18	17	72	501	10
45	6	38	16	15	29	669	10
46	2	48	1	1	458	685	10
47	8	56	27	26	34	362	10
48	13	52	36	35	694	962	10
49	6	68	30	29	444	597	10
50	47	47	13	12	499	690	10

$i$	$x_i$	$y_i$	$p_i$	$v_i$	$e_i$	$l_i$	$s_i$
51	49	58	10	9	296	505	10
52	27	43	9	8	111	362	10
53	37	31	14	13	301	602	10
54	57	29	18	17	595	780	10
55	63	23	2	1	442	917	10
56	53	12	6	5	426	851	10
57	32	12	7	6	366	585	10
58	36	26	18	17	458	980	10
59	21	24	28	27	17	441	10
60	17	34	3	2	651	932	10
61	12	24	13	12	246	421	10
62	24	58	19	18	25	583	10
63	27	69	10	9	34	348	10
64	15	77	9	8	46	523	10
65	62	77	20	19	49	608	10
66	49	73	25	24	440	845	10
67	67	5	25	24	132	563	10
68	56	39	36	35	464	925	10
69	37	47	6	5	12	535	10
70	37	56	5	4	249	968	10
71	57	68	15	14	218	417	10
72	47	16	25	24	22	377	10
73	44	17	9	8	263	444	10
74	46	13	8	7	479	965	10
75	49	11	18	17	182	399	10
76	49	42	13	12	166	497	10
77	53	43	14	13	430	970	10
78	61	52	3	2	212	671	10
79	57	48	23	22	273	580	10
80	56	37	6	5	452	968	10
81	55	54	26	25	388	479	10
82	15	47	16	15	99	352	10
83	14	37	11	10	21	328	10
84	11	31	7	6	323	624	10
85	16	22	41	40	333	504	10
86	4	18	35	34	350	503	10
87	28	18	26	25	380	487	10
88	26	52	9	8	114	553	10
89	26	35	15	14	649	981	10
90	31	67	3	2	345	534	10
91	15	19	1	1	675	930	10
92	22	22	2	1	18	673	10
93	18	24	22	21	339	969	10
94	26	27	27	25	403	536	10
95	25	24	20	18	22	301	10
96	22	27	11	10	565	732	10
97	25	21	12	11	582	703	10
98	19	21	10	9	122	367	10
99	20	26	9	8	185	580	10
100	18	18	17	16	297	965	10

Tabla de vehículos.

$k$	$P_k$	$V_k$
1	250	200
2	150	100
3	150	120
4	350	300
5	400	300
6	450	350
7	750	700
8	700	600
9	995	950
10	995	960
11	990	950
12	850	800
13	750	700
14	500	400
15	550	450
16	600	550
17	700	650
18	400	350
19	350	300
20	400	300
21	450	350
22	750	700
23	700	600
24	995	950
25	995	960

Anexo 5. Mejor solución encontrada en la instancia R201

Cantidad de vehículos: 4. Distancia total recorrida: 1296.59

Ruta 1 – Código Vehículo: 16 Distancia recorrida: 372.55

Utilización (Peso, Volumen): (92.00%,86.67%)

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
0	0.00	0.00	0.00	0.00	0	0	1000
33	24.76	24.76	20.24	55.00	10	45	200
65	51.33	81.57	0.00	91.57	10	49	188
63	87.23	127.47	0.00	137.47	10	34	191
31	104.70	154.94	0.00	164.94	10	152	275
69	112.51	172.75	0.00	182.75	10	162	293
52	123.28	193.52	0.00	203.52	10	174	299
27	131.82	212.06	0.00	222.06	10	104	255
28	138.53	228.77	0.00	238.77	10	114	255
12	147.75	247.99	0.00	257.99	10	228	345
29	163.40	273.64	0.00	283.64	10	190	313
76	178.40	298.64	0.00	308.64	10	290	373
30	198.53	328.77	0.00	338.77	10	259	354
71	217.31	357.56	0.00	367.56	10	268	367
9	225.56	375.80	24.20	410.00	10	400	497
51	231.88	416.32	0.00	426.32	10	348	453
81	239.10	433.54	0.00	443.54	10	411	456
79	245.42	449.86	0.00	459.86	10	388	465
78	251.08	465.52	0.00	475.52	10	384	499
34	256.08	480.52	48.48	539.00	10	529	614
3	270.22	553.14	0.00	563.14	10	527	584
50	278.47	571.39	0.00	581.39	10	547	642
68	290.51	593.43	43.57	647.00	10	637	752
54	300.56	657.05	0.00	667.05	10	641	734
4	309.78	676.27	1.73	688.00	10	678	801
55	318.32	696.54	0.00	706.54	10	620	739
25	321.93	710.15	106.85	827.00	10	817	956
24	336.93	842.00	0.00	852.00	10	704	847
80	346.15	861.22	0.00	871.22	10	839	968
77	352.85	877.93	0.00	887.93	10	817	952
0	372.55	907.63	0.00	0.00	0	0	1000

Ruta 2 – Código Vehículo: 17      Distancia recorrida: 273.74  
Utilización (Peso, Volumen): (84.22%,68.00%)

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
0	0.00	0.00	0.00	0.00	0	0	1000
95	14.87	14.87	77.13	102.00	10	92	231
59	18.87	106.00	0.00	116.00	10	17	229
92	21.10	118.24	0.00	128.24	10	18	181
42	31.30	138.43	0.00	148.43	10	25	172
15	40.52	157.65	17.35	185.00	10	175	300
14	56.33	200.81	0.00	210.81	10	32	243
44	61.99	216.47	16.53	243.00	10	233	340
98	72.62	253.63	0.00	263.63	10	183	306
61	80.23	271.25	18.75	300.00	10	290	377
16	84.71	304.47	0.00	314.47	10	272	373
38	100.52	330.28	0.00	340.28	10	302	405
86	113.56	353.32	34.68	398.00	10	388	465
85	126.20	410.65	0.00	420.65	10	376	461
99	131.86	426.31	0.00	436.31	10	333	432
94	137.94	442.39	0.00	452.39	10	436	503
6	141.11	455.55	0.00	465.55	10	415	514
53	153.15	477.59	0.00	487.59	10	414	489
26	161.21	495.65	92.35	598.00	10	588	667
56	180.91	617.70	0.00	627.70	10	585	692
74	187.98	634.77	46.23	691.00	10	681	802
91	219.55	722.58	16.42	749.00	10	739	866
100	222.72	752.16	45.84	808.00	10	798	965
93	228.72	814.00	0.00	824.00	10	811	969
17	243.03	838.32	0.00	848.32	10	733	870
60	255.68	860.97	0.00	870.97	10	721	862
89	264.74	880.02	0.00	890.02	10	755	920
0	273.74	899.02	0.00	0.00	0	0	1000

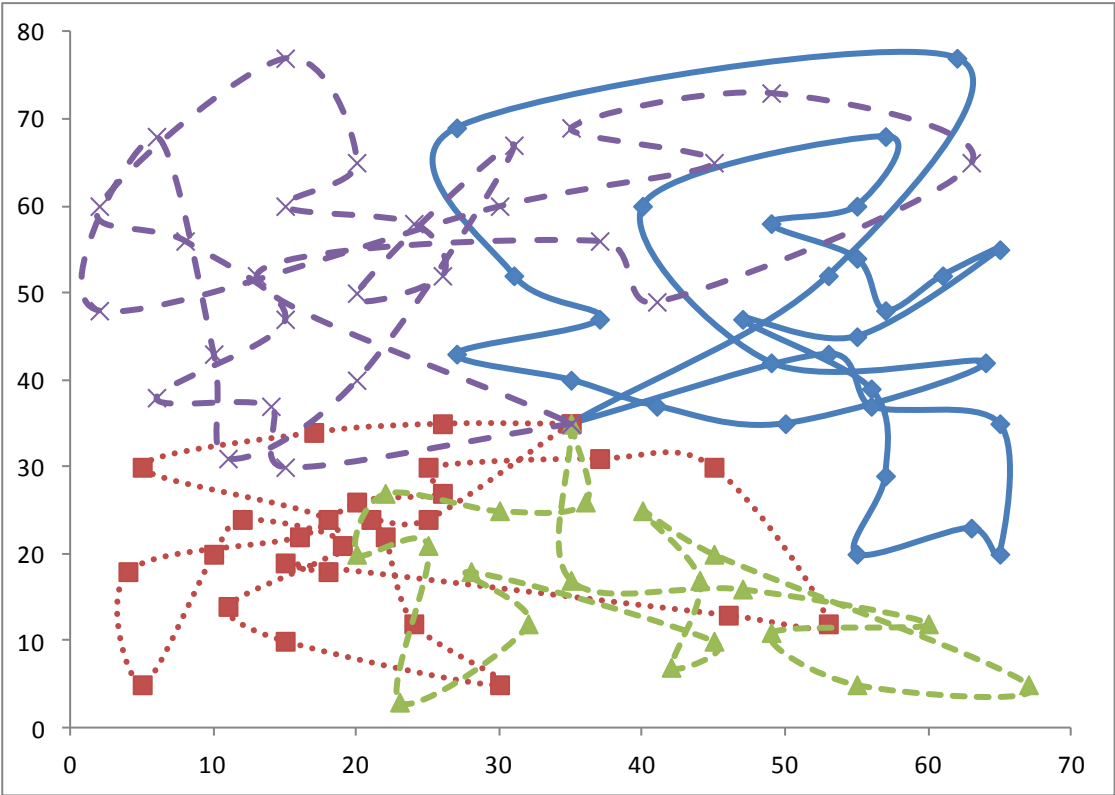
Ruta 3 – Código Vehículo: 21      Distancia recorrida: 224.86  
 Utilización (Peso, Volumen): (99.67%,89.00%)

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
0	0.00	0.00	0.00	0.00	0	0	1000
2	18.00	18.00	125.00	153.00	10	143	282
72	30.04	165.04	0.00	175.04	10	31	208
39	43.64	188.64	0.00	198.64	10	33	224
75	54.69	209.69	26.31	246.00	10	236	345
23	63.17	254.49	0.00	264.49	10	206	325
67	75.17	276.49	17.51	304.00	10	294	401
21	101.80	330.63	0.00	340.63	10	214	331
40	108.87	347.70	12.30	370.00	10	360	437
73	117.82	378.94	0.00	388.94	10	308	399
41	128.01	399.14	0.00	409.14	10	396	511
22	132.26	413.38	0.00	423.38	10	409	494
87	151.05	442.17	0.00	452.17	10	420	447
57	158.26	459.38	0.00	469.38	10	421	530
43	170.98	482.11	137.89	630.00	10	620	705
97	189.09	648.11	0.00	658.11	10	612	673
37	194.19	663.21	0.00	673.21	10	606	693
96	201.47	680.49	0.00	690.49	10	607	690
13	209.72	698.74	0.00	708.74	10	690	827
58	215.80	714.82	134.18	859.00	10	849	980
0	224.86	868.06	0.00	0.00	0	0	1000

Ruta 4 – Código Vehículo: 18      Distancia recorrida: 425.45  
Utilización (Peso, Volumen): (81.33%,79.33%)

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
0	0.00	0.00	0.00	0.00	10	0	1000
5	20.62	20.62	13.38	44.00	10	34	209
83	27.69	51.07	44.93	106.00	10	96	249
45	35.75	114.06	0.00	124.06	10	29	189
82	48.48	136.79	25.21	172.00	10	162	289
47	59.88	183.40	0.00	193.40	10	85	250
36	67.09	200.61	0.00	210.61	10	41	208
64	88.49	232.01	0.00	242.01	10	214	333
11	101.49	255.01	0.00	265.01	10	206	325
19	108.56	272.08	0.00	282.08	10	269	378
62	117.78	291.30	0.00	301.30	10	163	302
88	124.11	307.63	0.00	317.63	10	279	388
7	130.43	323.95	7.05	341.00	10	331	410
90	150.68	361.25	30.75	402.00	10	392	487
18	179.83	431.15	0.00	441.15	10	377	434
84	192.56	453.88	0.00	463.88	10	436	511
8	204.60	475.92	0.00	485.92	10	404	481
49	229.92	511.24	0.00	521.24	10	501	540
46	250.32	541.64	0.00	551.64	10	515	628
10	280.78	582.10	0.00	592.10	10	577	632
20	296.59	607.91	0.00	617.91	10	581	666
32	307.36	628.68	31.32	670.00	10	660	777
66	321.92	684.56	0.00	694.56	10	592	693
35	338.05	710.68	0.00	720.68	10	686	813
1	365.25	747.89	0.00	757.89	10	707	848
70	373.31	765.95	22.05	798.00	10	788	968
48	397.64	822.33	0.00	832.33	10	773	906
0	425.45	860.13	0.00	0.00	0	0	1000

Grafica de rutas instancia R201





Anexo 6. Mejor solución encontrada en la instancia R209

Cantidad de vehículos: 3. Distancia total recorrida: 969.98

Ruta 1 – Código Vehículo: 16 Distancia recorrida: 283.83

Utilización (Peso, Volumen): (86.50%,88.55%)

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
0	0.00	0.00	0.00	0.00	0	0	1000
95	14.87	14.87	7.13	32.00	10	22	301
2	27.07	44.21	29.79	84.00	10	74	351
39	52.57	109.50	0.00	119.50	10	33	797
67	62.47	129.39	2.61	142.00	10	132	563
23	74.47	154.00	0.00	164.00	10	36	513
75	82.95	172.49	9.51	192.00	10	182	399
72	88.34	197.39	0.00	207.39	10	22	377
21	92.81	211.86	0.00	221.86	10	156	389
73	95.97	225.02	37.98	273.00	10	263	444
22	103.04	280.07	0.00	290.07	10	280	623
41	107.29	294.31	0.00	304.31	10	225	682
98	134.21	331.24	0.00	341.24	10	122	367
85	137.37	344.40	0.00	354.40	10	333	504
61	141.85	358.87	0.00	368.87	10	246	421
16	146.32	373.35	0.00	383.35	10	222	423
86	152.64	389.67	0.00	399.67	10	350	503
44	160.70	407.73	0.00	417.73	10	72	501
38	171.52	428.55	0.00	438.55	10	147	560
14	182.70	449.73	0.00	459.73	10	32	454
87	197.97	474.99	0.00	484.99	10	380	487
57	205.18	492.21	0.00	502.21	10	366	585
15	212.46	509.49	0.00	519.49	10	30	529
43	219.74	526.77	51.23	588.00	10	578	747
42	228.79	597.06	0.00	607.06	10	25	616
97	237.85	616.11	0.00	626.11	10	582	703
92	241.01	629.27	0.00	639.27	10	18	673
100	246.67	644.93	0.00	654.93	10	297	965
91	249.83	658.09	16.91	685.00	10	675	930
37	254.93	690.10	0.00	700.10	10	563	736
93	259.40	704.57	0.00	714.57	10	339	969
96	264.40	719.57	0.00	729.57	10	565	732
13	272.65	737.82	0.00	747.82	10	426	978

0	283.83	759.00	0.00	0.00	0	0	1000
---	--------	--------	------	------	---	---	------

Ruta 2 – Código Vehículo: 23 Distancia recorrida: 361.38

Utilización (Peso, Volumen): (75.43%,81.33%)

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
0	0.00	0.00	0.00	0.00	0	0	1000
27	5.00	5.00	0.00	15.00	10	5	612
52	13.54	23.54	87.46	121.00	10	111	362
31	23.39	130.85	0.00	140.85	10	91	336
88	28.39	145.85	0.00	155.85	10	114	553
62	34.72	162.17	0.00	172.17	10	25	583
63	46.12	183.58	0.00	193.58	10	34	348
64	60.54	208.00	0.00	218.00	10	46	523
11	73.54	231.00	0.00	241.00	10	33	510
19	80.61	248.07	0.00	258.07	10	215	432
47	88.67	266.13	0.00	276.13	10	34	362
7	102.09	289.55	2.45	302.00	10	292	449
82	107.92	307.83	0.00	317.83	10	99	352
83	117.97	327.88	0.00	337.88	10	21	328
5	125.04	344.95	0.00	354.95	10	20	370
99	131.45	361.36	0.00	371.36	10	185	580
59	133.68	373.59	0.00	383.59	10	17	441
94	139.51	389.42	13.58	413.00	10	403	536
6	142.68	416.16	0.00	426.16	10	266	663
18	153.86	437.34	0.00	447.34	10	349	462
84	166.58	460.07	0.00	470.07	10	323	624
17	172.67	476.15	0.00	486.15	10	409	959
45	180.73	494.22	0.00	504.22	10	29	669
8	187.13	510.62	0.00	520.62	10	288	597
46	196.57	530.05	0.00	540.05	10	458	685
36	208.57	552.05	0.00	562.05	10	41	710
49	217.51	571.00	0.00	581.00	10	444	597
10	242.81	606.30	0.00	616.30	10	496	713
70	250.87	624.36	0.00	634.36	10	249	968
1	258.93	642.42	0.00	652.42	10	636	919
50	265.26	658.74	0.00	668.74	10	499	690
68	277.30	680.79	0.00	690.79	10	464	925
80	279.30	692.79	0.00	702.79	10	452	968
24	288.52	712.01	0.00	722.01	10	633	918
54	298.52	732.01	0.00	742.01	10	595	780

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
25	310.61	764.10	0.00	774.10	10	400	956
4	320.61	784.10	0.00	794.10	10	470	965
56	328.86	802.34	0.00	812.34	10	426	851
74	335.93	819.41	0.00	829.41	10	479	965
58	352.33	845.81	0.00	855.81	10	458	980
0	361.38	864.87	0.00	0.00	0	0	1000

55	307.00	750.49	0.00	760.49	10	442	917
----	--------	--------	------	--------	----	-----	-----

Ruta 3 – Código Vehículo: 14      Distancia recorrida: 324.77

Utilización (Peso, Volumen): (82.20%,95.75%)

$i$	$t_{ij}$	$a_i$	$w_i$	$d_i$	$s_i$	$e_i$	$l_i$
0	0.00	0.00	0.00	0.00	0	0	1000
28	6.32	6.32	37.68	54.00	10	44	325
12	15.54	63.22	106.78	180.00	10	170	403
76	22.62	187.07	0.00	197.07	10	166	497
29	37.62	212.07	0.00	222.07	10	129	374
79	46.83	231.29	41.71	283.00	10	273	580
65	76.26	312.43	0.00	322.43	10	49	608
71	86.56	332.72	0.00	342.72	10	218	417
9	94.80	350.97	0.00	360.97	10	254	643
81	100.80	366.97	21.03	398.00	10	388	479
33	103.63	400.83	0.00	410.83	10	24	646
51	110.84	418.04	0.00	428.04	10	296	505
30	120.06	437.26	0.00	447.26	10	115	498
90	131.47	458.66	0.00	468.66	10	345	534
69	152.35	489.54	0.00	499.54	10	12	535
53	168.35	515.54	0.00	525.54	10	301	602
40	175.05	532.25	0.00	542.25	10	243	554
26	182.13	549.32	0.00	559.32	10	548	707
77	197.39	574.59	0.00	584.59	10	430	970
3	200.22	587.41	0.00	597.41	10	498	613
78	209.44	606.63	0.00	616.63	10	212	671
34	214.44	621.63	0.00	631.63	10	487	656
35	224.64	641.83	0.00	651.83	10	439	948
66	240.76	667.96	0.00	677.96	10	440	845
20	249.70	686.90	0.00	696.90	10	538	709
32	260.47	707.67	0.00	717.67	10	602	835
48	288.28	745.47	0.00	755.47	10	694	962
60	306.72	773.91	0.00	783.91	10	651	932
89	315.77	792.97	0.00	802.97	10	649	981
0	324.77	811.97	0.00	0.00	0	0	1000

Grafica de ruta instancia R209

